An Unmanned Aerial Vehicle for Autonomous Forestry Cutover Edge Survey

David Hunt

A thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Engineering
in
Electrical and Computer Engineering
at the
University of Canterbury,
Christchurch, New Zealand.

29 February 2016

ABSTRACT

A proof of concept unmanned aerial system capable of navigating the edge defined by the boundary of forest and cutover has been developed. Following gathering of low altitude imagery of forest cutover edges, two image classifiers were developed to classify regions of an image as forest or not-forest from RGB images. The first proposed method, based on selection of hue, saturation, and brightness, achieved a classification accuracy of 83.2%. The second proposed method, based on a SVM classifier, achieved a higher classification of 90.2% by using textural information to avoid false positives.

The two methods were combined in order to accurately identify the cutover edge and fit a linear line. A proportional control system was used to generate navigational instructions based on the position and orientation of the line within a frame from a downward facing camera onboard a UAV. Mavlink commands were generated on an external computer onboard the UAV and sent to a Pixhawk flight computer to control a UAV. Successful performance of the proposed combined cutover edge identification algorithm was demonstrated in simulation with an average error of 26.2 pixels, approximately corresponding to a metric value of only 2.4 m. The simulated accuracy of this system compares favourably to a similar automated UAV system (Rathinam et al. [2007]) whose average positional error was 7 m.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would first like to thank my supervisors Kelvin Barnsdale and Dr. Richard Green who have provided guidance and support throughout the duration of this project. Their collective expertise and advice regarding electronics, UAV's, and computer vision proved extremely helpful throughout the year. Kelvin's UAV piloting skill (and UAV repairing skill) deserves special acknowledgment for allowing actual flight testing of the system. Richard's willingness to provide feedback, insight, and creative solutions during research and writing of this thesis has been greatly appreciated.

This thesis would not have been possible without the financial support and industry experience provided by SCION, in particular Bryan Graham and his colleagues. Bryan's part in fortnightly meetings provided essential insight into the nature of cutover edge, and his team's willingness and ability to provide supplementary video footage of cutover was invaluable.

I would also like to acknowledge forestry companies Nelson Forests and Rayonier for their assistance; Nelson forests for providing UAV imagery of cutover and Rayonier for providing access to active local forestry operations.

I offer special thanks to my fellow post graduate students Ryan Estep and Kieran Morris who were extremely helpful with brainstorming and helping me with my learning to use C++, makefiles, and Linux. Making the jump from Mechanical Engineering with limited programming experience was challenging but eased by your help.

Finally, I wish to thank my family for their ongoing support and encouragement during the course of my studies.

NOTATION

The following acronyms and definitions are commonly used in this thesis:

- UAV - Unmanned Aerial Vehicle
- UAS - Unmanned Aerial System(s)
- ALS - Airborne Laser Scanning
- HSV - Hue, Saturation, Value
- RGB - Red, Green, Blue
- IMU - Inertial Measurement Unit
- GPS - Global Positioning System
- DGPS - Differential Global Positioning System
- SfM - Structure from Motion
- DEM - Digital Elevation Model
- Orthomosiac - composite image sewn together from multiple images

Chapter 1

INTRODUCTION & MOTIVATION

Scion, a New Zealand Crown Research Institute (CRI), is investigating the use of UAV technology in a wide variety of applications for the forestry industry. Scion specialise in research, science, and technology development for the forestry, wood product and derived materials, and other bio-material sectors. This thesis is part Scion's investigation into emerging UAV technologies aiming to improve efficiency of harvesting operations.

In the forestry industry, the area where trees from the plantation have been felled is known as the 'cutover'. Location and area of the cutover are useful to forestry operators as it directly relates to the logging contractors remuneration. Actual cutover is currently measured infrequently, averaging 4-6 times a year. Cutover survey is performed using expensive satellite imagery, survey aircraft, or walking of the edge with an accurate GPS module. If aerial imagery is used then manual identification of the edge is performed. The manually identified edge is then fed into a software package for forestry monitoring. UAVs present an opportunity to dramatically improve this process in several ways. Firstly, it is anticipated that accuracy of the cutover edge measurements will be able to be increased using a low flying UAV with standard GPS compared to satellite or aerial imagery; the resolution of current cutover imagery is approximately 17 metres per pixel (Graham [2015]), so a low flying UAV will be able to capture images with a much greater spatial resolution. Secondly, use of a UAV for cutover survey will allow for the edge to be surveyed far more regularly, due to the low operational cost of a UAV, compared to acquisition of satellite imagery, or an aerial survey flight. The low cost of the operation also allows for much more frequent readings of the cutover location, the client expects that using a UAV for cutover survey approximately once per week. The increased temporal and spatial resolution of data acquisition allows for significantly closer monitoring of a contractor's progress. Finally a UAV cutover survey system has potential to integrate additional features such as monitoring of cutover condition, or checking trees for disease - other useful information for forestry operators.

This Masters project with Scion aims to develop a proof-of-concept UAV platform that can autonomously record the cutover edge, improving on current methods of cutover detection terms of frequency, accuracy, and cost.

1.1 OBJECTIVES

The purpose of this thesis is to develop a proof-of-concept UAV platform which can be used with minimal operator input to identify the forest cutover edge with high reliability. The cutover edge shall be recorded as a series of GPS coordinates, which may be later imported into the forestry company's GIS tools for managing the forestry operations. The specific objectives for the project are:

- Identify appropriate methods and technologies which may be suitable for the detection of, and navigation through forests,
- Meet with Scion representative to identify specifications of the UAV system,
- Collect sample data to facilitate future development,
- Develop edge detection and navigation algorithms for cutover edge detection using sample data to validate,
- Test algorithms on sample data,
- Port edge detection algorithms to embedded solution for placing on board UAV,
- Integrate embedded system into UAV platform and control system,
- Perform testing of integrated system in a simulation,
- Perform flight trials of proof-of-concept platform.

1.2 CONTEXT AND CHALLENGES

Forestry is a significant industry in New Zealand, exotic plantation forests cover 1.75 million hectares, about 7% of New Zealand's land surface area. The industry contributes an annual gross income of approximately $5 billion, 3% of New Zealand's GDP. The plantations are 90% Radiata pine (Pinus radiata), 6% Douglas-fir (Pseudotsuga menziesii), with the remainder made up of eucalyptus and other species. Fully grown, the forests are approximately 30 m - 35 m tall (Graham [2015]). The cutover edge of forests can be up to 10 km long. Such a distance is plausible to fly with unmanned UAVs, however to be feasible such situations would require the UAV flying beyond visual line of site. This gives rise to many engineering issues regarding range of telemetry, video, and radio control feedback over large distances, but also a legal challenge considering the current legal framework surrounding use of UAVs. The legal aspect has been examined briefly in Appendix A. Due to the scope of this thesis, the legality and engineering issues associated with flight beyond visual line of sight are not relevant, though this is an area that will be important in the future work related to this thesis.

Another challenge on this project is ensuring a high degree of operational safety while operating with minimal user input. To achieve this rigorous testing of different aspects of the system will be required to validate navigational decisions made. New Zealand's Forests are often located in undulating, or even mountainous terrain. Such terrain complicates the detection and navigation tasks as a safe operating height above the canopy must be maintained, and steep terrain could cause casting of shadows across some areas of the forest and not others. Techniques and technologies enabling consistent height above ground have been explored in Chapter 2.

1.3 CONTRIBUTIONS

The primary contribution of this thesis is a cutover edge detecting algorithm based on colour selection and SVM trained classification (Section 3.2). The algorithm operates onboard a UAV and fits a linear line corresponding to the cutover edge location to sequential video frames taken from a downward facing camera at approximately 10Hz. Performance of the system has been validated offline using multiple videos taken of a cutover edge from a UAV. The algorithm developed is novel compared to similar systems examined in Section 2.3 as it operates in real-time on low altitude imagery.

The system in which the algorithm is a novel contribution to forestry research as it makes navigation decisions in real time based upon the output of the cutover detection algorithm. Current UAV systems in use in the forestry sector do not have any level of autonomous navigation, rather relying on a preplanned series of way points for mapping of forest areas. Allowing the system to make navigational decisions has many potential benefits such as reduced pilot involvement and expertise required, greater length of cutover edge coverable in single flight using a preplanned grid, and potential for beyond line of sight capabilities.

1.4 STRUCTURE

This thesis begins with a literature review, which covers the state-of-the-art in related research areas and background required for the proposed methods described in this thesis. Chapter 3 describes the proposed algorithms for forest edge identification and navigation. Chapter 4 describes the hardware used, implementation, as well as the overall system design. The software, hardware, and testing of the proposed algorithms and UAV system is described in Chapter 5, and project conclusions as well as potential research directions are discussed in Chapter 6. Appendices include guides for recreating the system setup.

Chapter 2

TECHNOLOGY REVIEW AND RELATED WORK

In recent years the popularity and accessibility of Unmanned Aerial Vehicles (UAVs) has increased significantly. The emergence of affordable UAVs has led to the realisation of many innovative civil applications. UAVs have been used for a survey of the archaeological site of Pinchango Alto by Eisenbeiss et al. [2005], where rugged topography made access and working on the site difficult. Use of UAVs in disaster response has become a reality, Pratt et al. [2006] explored use of UAVs for inspection of commercial buildings damaged during Hurricane Katrina and Chou et al. [2010] used a UAV for aerial survey of damaged caused by Typhoon Morakot in Taiwan where access was difficult due to floods, landslides, and infrastructure damage. These are just a few examples of potential UAV usage, other applications include search and rescue, law enforcement, monitoring of engineering sites, crop management in agriculture, assessment of air quality, and many others. It is apparent that due to their mobility and low cost relative to alternatives, UAVs have potential to improve a wide variety of civilian operations.

The task of forestry survey is becoming routine for UAV companies around the world, for example using footage from UAVs for volumetric surveys of processed wood quantities (mosaicmill.com [2015]), or offering mapping services (altus.com [2016], smartplanes.com [2015]), from which one can expect a referenced orthomosiac composite image showing the wider area, and a digital elevation map (DEM) showing a 3D reconstruction of the site . An example of this is Canadian company UAV Geomatics, who were commissioned in an investigation into the vulnerability of peatland to wildfire. UAV Geomatics were able to generate an orthomosiac and digital elevation model (DEM) by flying a grid pattern over the regrowing forest (Wanless [2013]). While these UAV companies present an alternative to traditional cutover survey techniques which require satellite or manned survey aircraft imagery, they are only a partial solution as they are required to fly a large distance in order to map a relatively small area due to the nature of flying a grid pattern. The large required flight distance reduces the length of cutover edge which can be mapped in a single flight, or from a single starting point. In addition, the generated orthomosiac or DEM needs may require extensive post processing to extract the edge information of the cutover.

This section reviews work relevant to the project goal of flying an unmanned aerial vehicle semi-autonomously along the cutover edge, a problem encompassing many different research area. Detecting the difference between cutover forest requires consideration of machine vision hardware, remote sensing, and image recognition techniques. Flying a UAV autonomously requires consideration of controls, safety, obstacle avoidance, and awareness of legal implications. To ensure readability of this section, the literature and technology review has been broken into the following sections for clarity:

1. An overview of imaging sensors and their suitability for this project,
2. Computer vision based forest edge segmentation algorithms,
3. Remote sensing of forestry areas,
4. UAV navigation and vision assisted examples,

2.1 IMAGING SENSORS

Potential technologies which were considered for detection of the cutover edge have been summarised in this section. Note that there is a particular emphasis in this section on acquisition of depth information as early in the project it was anticipated that identifying the sudden change in height from cutover to treetops would be a suitable way of quickly identifying the edge.

2.1.1 LIDAR

LIDAR is a combination of the words “Light” and “Radar” (OxfordDictionary [2015]). The LIDAR sensor calculates distances to objects by measuring the time of flight (TOF) taken for laser pulse to travel from the laser, to the object, and back to the sensor. Distance measurements can be collected in large quantities very quickly by combining multiple lasers and detectors in one sensor. By rotating multiple sensors at different orientations, distance information for entire 3D environments can be recorded and processed into accurate point clouds, an example of such is shown in Figure 2.1. LIDAR is used in a variety of fields including geology, seismology, forestry, and astronomy. Due to the use of lasers, typical characteristics of LIDAR include high precision, long range, large mass, and high cost. A beneficial feature of some LIDAR units is that they are capable of measuring not just the first return of the laser pulse, but also the second and third returns. The capability of measuring multiple return signals is particularly useful when flying over trees or other vegetation as it allows mapping of features below the canopy.

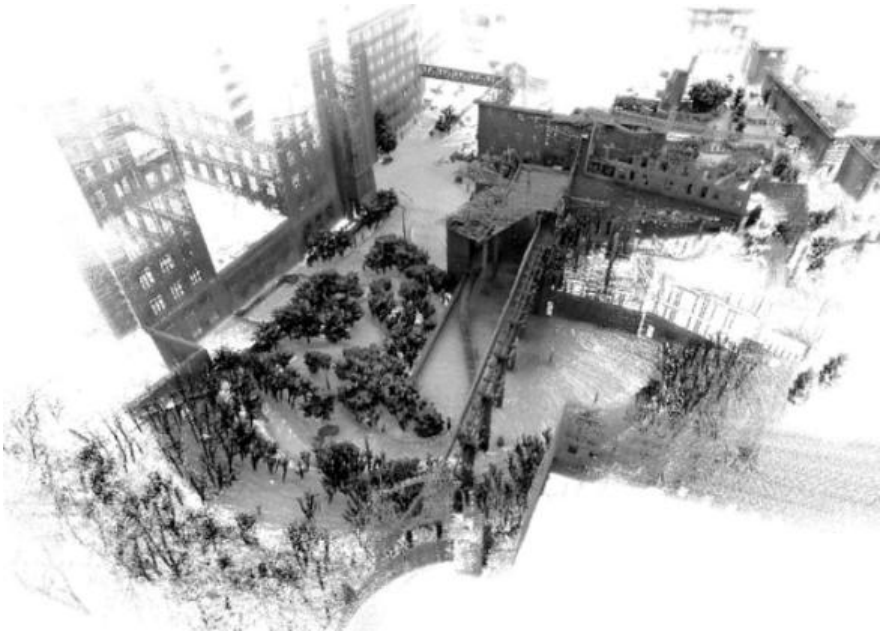


Figure 2.1 An example of a point cloud of an urban area from LIDAR data (Velodyne [2015])

The recent development of lightweight LIDAR units, shown in Table 2.1, and multi-rotor UAVs capable of carrying them has allowed for highly accurate mapping without requiring a traditional aircraft. The availability of these small LIDAR units is increasing: Espositoa et al. [2014] noted in May, 2014 that the Yellowscan was the lightest LIDAR system on the market, whereas it is shown in Table 2.1 that as of January 2016 there are three LIDAR units lighter than the Yellowscan, the Ibeo Lux, and both Velodyne LIDARs.

Table 2.1 Comparison of UAV suitable LIDAR units.

Manufacturer	Model	Weight (kg)	Range (m)	Power (W)	Scan Angle (degrees)
Velodyne Lidar	VLP-16	0.83	120 at 80%	8	360
Velodyne Lidar	HDL-32E	1.05	120 at 80%	12	360
Ibeo Systems	LUX 4	1.10	200 at 90%	8	-60/50
Riegl	VUX-1	3.65	920 at 60%	30-60	330
YellowScan	YellowScan	2.1	150 max.	20	100
RouteScene	UAV LidarPod	2.5	100 max.	28	360

Several companies are now selling UAVS with one of these LIDAR units capable mapping and survey services:

- In early 2016, New Zealand company Altus (altus.com [2016]) began offering LIDAR services using the Velodyne HDL-32E, and the Velodyne VLP-16.
- XactSense, a company from USA also uses the Velodyne HDL-32E.
- coptercraft.com [2016] UAVs use the Riegl VUX-1 UAV.
- Another company from the UK, UAV sell a UAV with the Routscene LidarPod, a

turnkey system based on the Velodyne HDL-32E which has additional features such as RTK GPS.

Using LIDAR for forestry monitoring is an increasing trend (Corona et al. [2012], Lim et al. [2003], Montaghi et al. [2013], Wulder et al. [2008]). Espositoa et al. [2014] and Hummel et al. [2011] point out that UAVs have potential to improve performance due to their ability to fly at low altitudes to gather dense data, as well as their lower cost compared with hiring of aerial survey craft and crew. Espositoa et al. [2014] and Wallace et al. [2012] both performed studies using UAVs fitted with LIDAR units and were able to measure individual tree height with 5% and 0.15 m error respectively. These two studies in particular show the efficacy of using UAV based LIDAR solution for cutover edge identification, however the current cost, weight, and processing restrictions associated with using a UAV based LIDAR unit mean that it was not investigated further.

2.1.2 Conventional Cameras

In this section, conventional cameras refers to the device common in society for capturing a 2D representation scene in the visual spectrum, specifically an imaging device which uses a photo diode image sensor. The photo diode is a P-N junction diode which is a passive sensor, converting photons from incoming light into a proportional number of electrons. There are two primary image sensor types which manage the output from a photo diode in different ways, a CCD image sensor stores and move the electrons as they are before later conversion to voltage, a CMOS sensor immediately converts the incoming electron charge to a voltage. There are many potential options for post processing images for extracting information such as depth, scene recognition, and object position. A more detailed review of image processing methods is included in Section 2.2.

2.1.2.1 Stereo vision

Stereo vision is the process of deducing 3D information of a scene based on input from multiple cameras. By comparing the relative positions of objects visible in both frames, depth information can be extracted. Stereo vision is very similar to the biological process of how humans and other animals with binocular vision perceive depth, known as stereopsis. The separation of eyes gives rise to disparities which are processed in the visual cortex to yield a 3D scene Howard and Rogers [1995]. In digital stereo vision two cameras are separated horizontally by a fixed distance or baseline. In order to extract the depth information both individual images need to corrected for distortion, rectified so that they both share a common plane, and then a disparity map generated based upon the separation of features in each image. The disparity map is directly proportional

to a 3D depth map, which can be produced when parameters of the cameras and their orientation relative to each other have been calibrated. Figure 2.2 shows an example depth map produced using a stereo vision setup.

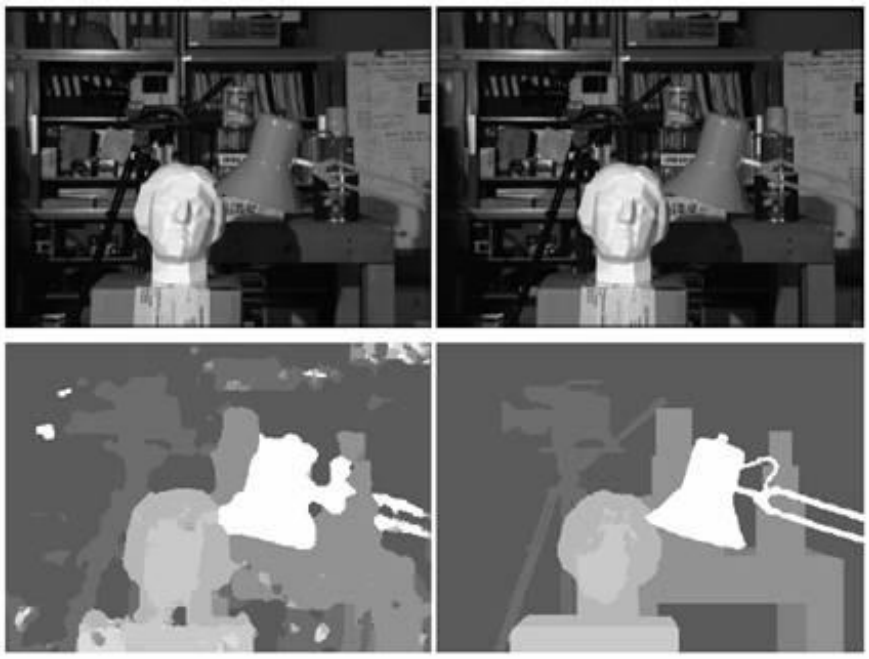


Figure 2.2 A stereo vision example from Domínguez-Morales et al. [2012] showing the left and right images (top), the ground truth depth (bottom right), and the computer generated depth map (bottom left)

The accuracy of stereo vision relies on a combination of several factors: the image resolution, distance from object, the camera focal length, and the distance between cameras (the baseline). The possible error associated with the estimation of a point (x, y, z) in space relative to the center of the image is described by the equation:

$$\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} = \frac{z\Delta d}{Bf}\sqrt{x^2 + y^2 + z^2}$$

As the baseline B and the focal length f are increased, the possible error decreases. The disparity error Δd, increases the error linearly, and the distance from the target increases the error quadratically. It is also important to note that the position within the image (x, y) also influences accuracy, the cente r of the image, where $x = y = 0$ is the most accurate.

Stereo vision has previously been used for mapping, navigation, and obstacle avoidance for unmanned ground vehicles (Van Der Mark et al. [2001], Matthies et al. [1996], Rankin et al. [2005]), and more unmanned aerial vehicles (Stefanik et al. [2011], Hrabar et al. [2005], Hrabar [2008], Byrne et al. [2006]). Hrabar [2008] used a forward facing stereo vision setup aboard a physical test rig for developing of object detection, and path planning. This was achieved by using the depth data from the stereo cameras

(a) Digital elevation map of a field next to a group of trees produced using stereovision.

(b) Stereovision rig mouted to Yamaha RMAX unmanned helicopter.

Figure 2.3 Stereovision rig and output from Stefanik et al. [2011]

to generate a point cloud, and further process the cloud into a 3D occupancy grid. An occupancy grid is made up of voxels, representations of a value in 3D space (like a pixel represents a value in 2D space) which are marked as occupied or unoccupied. The grid is then used to make autonomous navigation corrections in order to avoid occupied areas of space. Unfortunately, the system was not found to be reliable enough for implementation on to a physical system. Occupancy mapping and obstacle avoidance in general is however something that should be considered in future work as an added assurance of safe operation.

Stefanik et al. [2011] used a stereo vision setup on board a UAV to produce dense 3D terrain models in near real time. Accuracy of the system ranged from 56 cm to 65 cm across the field of view, from an altitude of 40 m. The stereo vision setup included a differential GPS unit (DGPS), and an independent inertial measurement unit (IMU) to determine the position and attitude of the cameras, bringing the total vision system to a total mass of 3.1 kg, lighter than LIDAR units available at the time. A resulting DEM and the test rig used is shown in Figure 2.3. Interestingly, Stefanik et al. [2011] performed a comparison between their stereo setup and a LIDAR unit to examine differences in accuracy with distance. As expected, it was was found that the error of the LIDAR unit remained consistent and small (less than 0.1 m), while the stereo vision error increased quadratically as distance from the target increased. Comparison of the two technologies highlights the major drawback of stereo vision; due to the inherent accuracy degradation with distance, such a system may not be suitable for the task faced in this thesis to the height of forests and hence the required height of the UAV above ground. Regardless, stereo vision shall be investigated further as accuracy of the system only needs to be sufficient to identify the difference between forest canopy, and the ground — an approximate height differential of 30 m.

2.1.2.2 Structure from motion

Another imaging technique for retaining depth information is the process known as structure from motion (SfM). SfM is a range imaging technique whereby the structure

of 3D objects and environments is inferred from a series of conventional 2D images (Ullman [1979]). The problem of estimating 3D structure from images is similar to that encountered in stereo vision where correspondences between images are required. SfM is however computationally more intensive as the displacement and attitude of the camera between images is not necessarily constant or known and there a larger number of images processed.

An example of a 3D surface produced from UAV imagery is found in Lucieer et al. [2013] where a SfM was used to generate a DEM and an orthomosiac of a landslide. Figure 2.4 shows the 3D surface produced. 98 images were taken of a landslide, GPS coordinates from the UAV controller were added to the image exif data to speed up processing. The processing time for the image feature detection and matching alone (performed in Photoscan on a desktop computer) was 75 minutes. Lucieer et al. also manually assessed the horizontal and vertical accuracy of the generated orthomosiac using 39 independent DGPS field reference discs and found the average accuracy was 6.1 cm with a standard deviation of 4.1 cm horizontally, and 0.6 cm with a standard deviation of 6.2 cm vertically, an accuracy definitely sufficient for the task explored in this thesis.

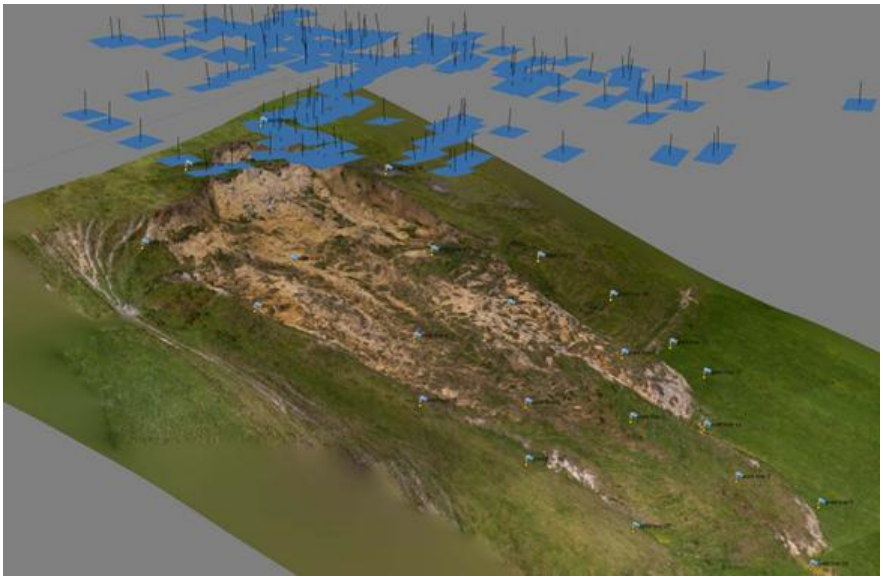


Figure 2.4 An image from Lucieer et al. [2013] showing a texture-mapped 3D surface of a landslide produced using SfM from UAV Imagery. The blue planes above the surface represent position and attitude of the camera at each image.

2.1.2.3 Optical Flow

Optical flow is a technique applied to video footage or sequential images to generate a two dimensional motion field from the projection of 3D velocities on onto the image plane. Optical flow algorithms originated in the 1980s (Horn and Schunck [1981], Lucas et al. [1981]). The concept of using optical flow to generate a motion field is similar to

the task of generating a depth map in stereovision âĂŞ multiple images are used to infer scene information based on differences between images. With optical flow the difference in the images is based on movement, either of the camera or of objects within the scene. Without movement between sequential frames, there is no useful information generated using optical flow methods. The 2D motion field is a useful feature that can be used for a variety of tasks such as image segmentation (Borş and Pitas [1998]), obstacle detection (Braillon et al. [2006], object tracking Mae et al. [1996]). Optical flow has also previously been used in a number of UAV related applications, such as landing assistance (Herisse et al. [2008]), stabilisation (Romero et al. [2009]), and obstacle avoidance during flight (Merrell et al. [2004]).
Optical flow would not be a complete solution to edge identification as it requires on movement of the camera or of the objects within the frame to generate useful information about the scene. This may be suitable on a fixed wing UAV which flies with a minimum forward velocity but not on a multi-rotor UAV capable of hovering. Combining optical flow with stereovision as has been previously been performed (Hrabar et al. [2005]) could be a solution to this problem.

2.1.3 Other Technologies

2.1.3.1 Interferometric Synthetic Aperture Radar

Interferometric synthetic aperture radar (InSAR) is a radar mapping technique. Multiple synthetic aperture radar (SAR) images can be used to generate digital delevation maps based on the phase difference of the radar waves (Allen [1995]). InSAR is typically used for remote sensing with data gathered from units situated on survey aircraft or satellites (Massonnet and Feigl [1998]). InSAR is not suitable for this project as a small UAV would not be able to lift the required equipment. Additionally the complexity of processing, cost, and accessibility of the technology compared to alternatives previously discussed also rule out use of InSAR.

2.1.3.2 Structured Light

Structured light is a technique involving the projection of a known pattern (in visual or infrared spectrum) and using the distortion of the pattern to infer information about the surface or object being scanned. (Scharstein and Szeliski [2003]). Products such as Intel RealSense and Microsoft Kinect utilise structured light to create depth maps. Stuctured light cameras and scanners are not appropriate for this project as they are currently only suitable for operating indoors, as bright sunlight saturates the projected infrared light pattern from the system.

2.1.3.3 Time of Flight

Time of flight (ToF) cameras operate on the same principle as LIDAR units: emit a pulse of light, detect the reflection, then use the time difference between transmission and receipt to determine depth information. Time of flight systems differ from LIDAR in that they illuminate the entire scene on each pulse, hence eliminating the need for rotating the transmitter and receiver. Unfortunately as a ToF camera is required to illuminate the entire scene with infrared, their usefulness outdoors and over long distances is limited due to poor signal to noise ratios, and so are not suitable for use on this project.

2.2 COMPUTER VISION BASED FOREST EDGE DETECTION

Processing of images to extract useful information is an active field of research. Recently advanced deep learning techniques have enabled companies Google and Microsoft to surpass the human benchmark in an image recognition challenge (He et al. [2015]). Similarly social media giant Facebook's 'DeepFace' project (Taigman et al. [2014]) is approaching human accuracy in the identification of faces. Applications of computer vision and image processing techniques are far reaching. Some examples include improved image searching, object quality inspection, motion sensing security cameras, cruise control in motor vehicles, and augmented reality, to name a few. This section begins with fundamental techniques and then explores more complex methods which may be suitable for identifying forestry from a UAV.

2.2.1 Colour Imaging

Colour is a useful feature for image processing as it is a powerful descriptor which can simplify segmentation and classification tasks. Colour processing can be classified as either full colour or false colour. In full colour processing the raw colour intensity information is used, while in false colour or pseudo colour processing the image is first transformed to grayscale using some criteria. RGB is a common colour model which represents a given colour as a combination of red, green, and blue intensity. There are however other colour models with their own advantages and disadvantages, the primary models are: RGB, CIE, YUV, HSL/HSV/HSI/HSB, and CMYK. The HSV colour space colour model represents colours as hue, saturation, and value/brightness (HSV or HSB), as shown in Figure 2.5. HSV is often favoured by designers as it allows for more intuitive selection of colours compared to RGB. For this reason the colour based forest detection method described in Section 3.2.1 loads an image, and subsequently transforms it from the from the RGB space to the HSV space. The SVM based forest

detector described in Section 3.2.2 does transform the images as it is based on a trained classifier where manual colour selection is not a requirement.

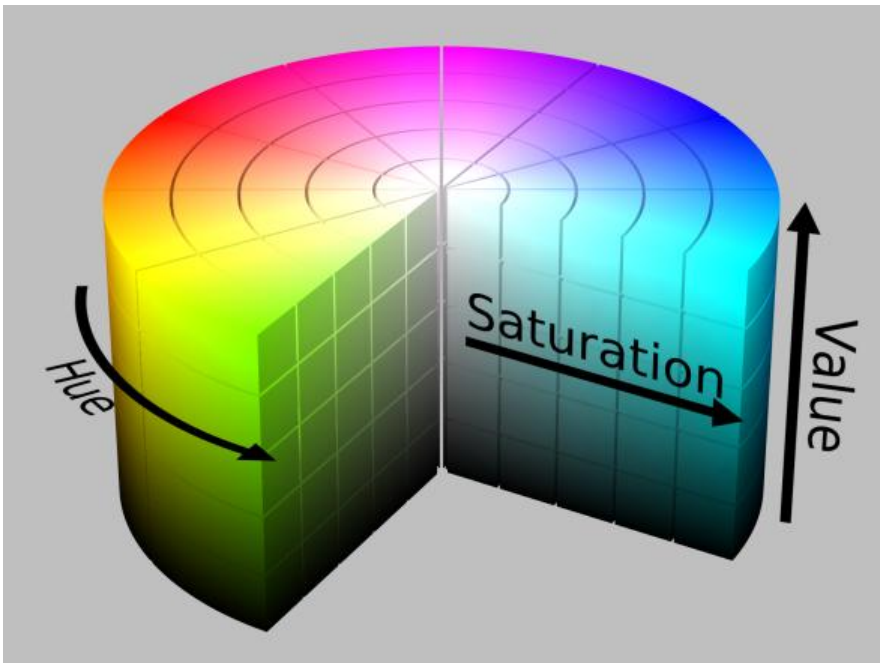


Figure 2.5 A graphical representation of the HSV colour space is shown. (wikipedia.com [2010])

2.2.1.1 Image Segmentation

Segmentation is the process of dividing an image in sections based on image contents, particularly relevant to this thesis as our proposed algorithms need to segment an image into areas of forest and cutover. Some applications of image segmentation are: analysis of medical imagery, face detection, pedestrian detection, fingerprint recognition, and traffic control systems. In some examples the aim is to segment an object of interest from the background, or to separate multiple classes or objects present in an image. "Image Processing, Analysis, and Machine Vision" Sonka et al. [2014], classifies segmentation techniques as falling into one of three categories, thresholding, edge based, or region based segmentation.

Thresholding

A simple, useful image manipulation is thresholding. Image thresholding involves separating an image into two classes based on their relation to a global threshold value; if a pixel is greater than the threshold then it is assigned one value, else if it is assigned another value. Thresholding is often used to turn greyscale images into binary black and white images to simplify future tasks.

There are a large number of more advanced thresholding techniques which do not use a single global threshold but rather adapt the threshold based on surrounding pixels; these techniques can be classified as adaptive thresholding techniques. Examples

of adaptive techniques are entropy based, shape based, locally adaptive, attribute similarity, clustering based, and histogram based thresholding. **??** shows some examples of results achieved using different methods.A comparative study (Sezgin et al. [2004]) found entropy based and clustering based thresholding methods to be the best methods for examining non-destructive testing images however it was noted that application specific information should be considered when selecting an appropriate thresholding method.

Edge Based Segmentation

Edges within images present an opportunity for separating areas into different segments. Edge based segmentation requires two steps, firstly the detection of likely edges, and secondly edge linking which is the combination of edges to form segments. While edge based segmentation methods are often simpler and faster than region based segmentation, they are susceptible to noise and are hence better suited for situation where crisp edges between segments are apparent.

Region Based Segmentation

Region based segmentation is the process of dividing up an image based on the properties of the regions themselves. Two common techniques are region splitting and merging, and region growing. In region splitting and merging, an image is divided into arbitrary regions which then merge or split further based on the properties of each region. Region growing starts of by placing seed points throughout the image, the seeds grow to form groups of similarly properties. One problem with region growing techniques is that it is difficult to tell the algorithm when to stop; one wants to segment the image until the region of interest (ROI) is separated from other regions but it is detrimental to break up the ROI into more than one region. Segmentation used region based techniques is effective and more robust to noise than edge based segmentation, however it requires more computation time due to the complexity of the growing and splitting/merging techniques.

2.2.1.2 Supervised Classification

Another potential way of segmenting an image is by applying a supervised classification method to separate classes based on image features. Sonka et al. [2014] classifies supervised classification as a separate task to segmentation, opting to categorise it as an object detection method. Supervised classification is not classed as an image segmentation method as it is a technique for finding one (or more) specific thing within an image. Supervised learning methods use some form of machine learning to use a priori information to 'learn' properties of the desired object or region. A classifier is an algorithm that assigns a class to an object based on some description of the object. Support vector machines (SVM), neural networks, and naive Bayes are all supervised learning methods.

Support Vector Machines (SVM)

A support vector machine is a statistical learning algorithm which is used for classification. A SVM uses training data which has been marked as belonging to a certain class to compute an optimal hyperplane separating into a discrete number of classes in high dimensional space. This can be thought of in a two dimension example as a line separating two types of points, as in Figure 2.6. The optimal separation hyperplane refers to the decision boundary which minimises false classifications in the training set. SVMs have been used in remote sensing for classifying forestry areas, and have been useful due to their ability to generalise well even with limited training data.

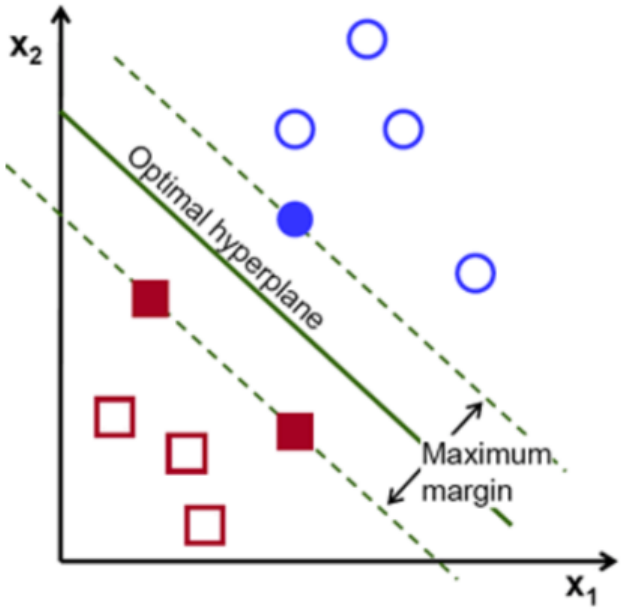


Figure 2.6 An example (Bradski) illustrating an optimal separating hyperplane. In this two dimensional example, the hyperplane is visible as a straight line.

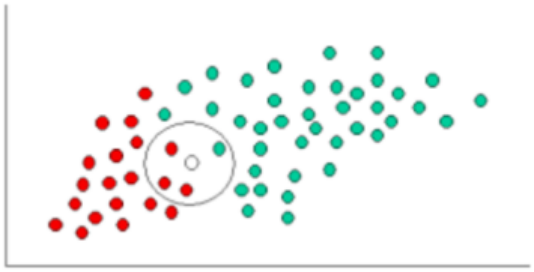


Figure 2.7 A classification example problem, would the white circle be better classified as red or green?

Naive Bayes Classifiers

Naive Bayes classifiers are based on the so-called Bayesian theorem. The theorem describes the probability of an event based on conditions related to the event. P(A|B) is the probability that of observing A given B. P(B|A) is the probability of observing B

given that A is true. P(B) and P(A) are the independent probabilities of A and B being served. Classification using the Naive Bayes method uses a priori information about the statistical distribution of different classes to formulate an expected probability for an unclassified point. For example if we were to try and classify the white point in Figure 2.7 as green or red we would calculate the Prior probability for green and red based on their current population sizes:

$$\begin{aligned} Prior\,probability\,for\,GREEN &= \frac{GREEN}{TotalNumber} &&= \frac{40}{60} \\ Prior\,probability\,for\,RED &= \frac{RED}{TotalNumber} &&= \frac{20}{60} \end{aligned}$$

Then the likelihood is calculated:

$$\begin{aligned} Likelihood\,of\,X\,given\,GREEN &= \frac{GREEN\,near\,X}{GREEN} &&= \frac{1}{40} \\ Likelihood\,of\,X\,given\,RED &= \frac{RED\,near\,X}{RED} &&= \frac{3}{20} \end{aligned}$$

The final classification is then produced by combining the prior information and the likelihood previously calculated:

$$\begin{aligned} Posterior\,probability\,X\,is\,GREEN &= Prior\,proability\,GREEN \times Likelihood\,given\,GREEN \\ &= \frac{40}{60} \times \frac{1}{40} \\ &= \frac{1}{60} \\ Posterior\,probability\,X\,is\,RED &= Prior\,proability\,RED \times Likelihood\,given\,RED \\ &= \frac{20}{60} \times \frac{3}{20} \\ &= \frac{1}{20} \end{aligned}$$

The white circle is hence classified as the posterior probability given red is higher. This is a demonstrative example however it shows the principle of classification using Naive Bayes.

Artificial Neural Networks

Artificial neural networks (ANN) are a biologically inspired family of models for machine learning. ANNs simulate the way that a human's brain functions, with interconnected neurons being used to learn from observation. An artificial network can have anything

from a few, to a few million interconnected neurons arranged in layers. **??** shows an example layout of neurons in an ANN. The input layer represents inputs values, such as a descriptor from an image feature. The middle layers are hidden layers, nodes in the hidden layers have individual weights which respond to different inputs values to eventually produce the output layer âĂŞ the desired answer. In our example of image classification the output might simply be a binary output representing whether the input feature descriptor was characteristic of a forest (or whatever the network was trained to detect) or not. Neurons receive multiple inputs from the input side of the network, and generate a scalar output which is fed onward though network to one or more neurons. Neurons have a certain activation level defined by an activation rule which acts on input signals to produce a new output signal. Weighted interconnections between neurons determine how activation of one neuron leads to input for another. Connection's weights are not initially able to manipulate the input signal to reach the desired output, they are generally randomly seeded and must be 'learn' a suitable weight. Learning in this context means modification of weights to adjust the input to output relationship. Figure 2.8 shows an simple 3 layer ANN to demonstrate the layout of nodes, layers, inputs, and outputs.

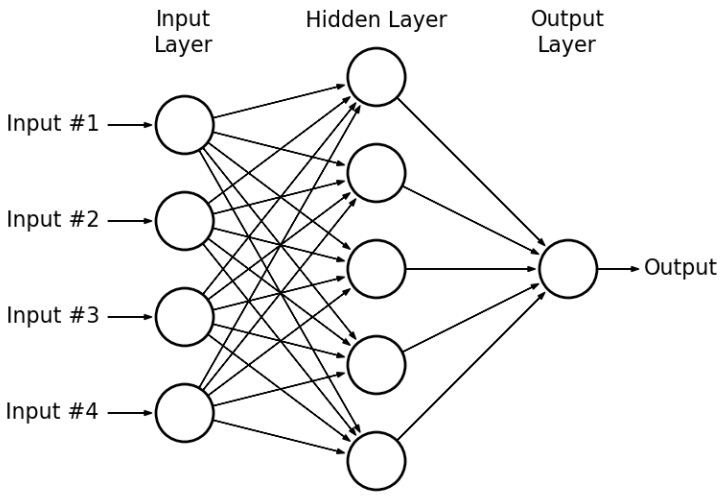


Figure 2.8 A simple 3 layer neural network with four inputs and one output.

ANNs are useful as they can evolve to sort complex patterns, where relationships between inputs and outputs are variable and non-linear. Alternative techniques are often limited by mathematical assumptions of variable independence, normality, or linearity. An ANN can capture a range of complex relationships which may be otherwise complex to express. The versatility of ANNs does come at a cost, they are complex and require long training times.

2.3 REMOTE SENSING OF FORESTRY FROM AERIAL IMAGERY

Remote sensing is a field of geography referring to the use of aerial or orbital sensor technologies to examine objects on Earth. Sensors used are either classified as active, where a signal is transmitted and reflected from earth (e.g. LIDAR), or passive, where a sensor responds to ambient light (commonly sunlight). Remote sensing is used in a variety of applications such as monitoring shoreline changes, ocean temperatures and circulation, tracking sea ice movement, and forestry analysis. Remote sensing images are typically digital and require some degree of image processing in order to extract useful information or to assist in visual interpretation of images. Segmentation and classification algorithms, such as those discussed in Section 2.2 are commonly used for classifying images into classes. An example of this is shown in Figure 2.9 from Lane et al. [2014] where a combination of classification techniques are used to classify components of a river delta.

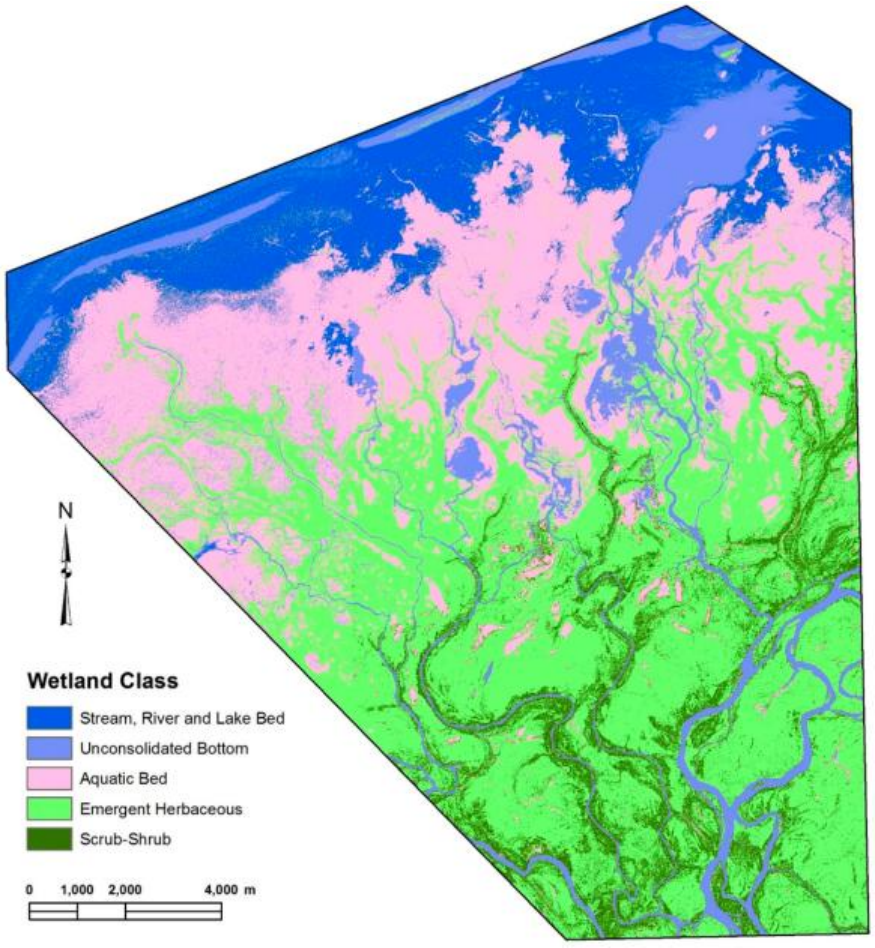


Figure 2.9 An example of image classification from Lane et al. [2014]. Satellite imagery was classified using a novel hybrid approach using supervised and unsupervised techniques.

Remote sensing has previously been used for the extraction of information on the forest structure. LIDAR is a common method due to the high spatial resolution data and vertical accuracies possible. As the application of LIDAR to forestry has been discussed in Section 2.1.1 this section will focus on other methods. Remote sensing often uses sensors which detect light beyond the visual spectrum to assist with image processing. A multispectral image is one where different bands represent the intensity of different wavelengths of light. Multispectral images commonly include the red, green, and blue bands we are familiar with, with one or more additional bands the infrared or ultraviolet spectrum. For example, the Landsat-8 observational satellite include has 11 bands in total, ranging from $0.43\mu m$ to $12.51\mu m$ Survey [2015]. The rear-infrared bands are particularly useful for imaging of vegetation due to the reflectivity properties of living plants. Photosynthesizing plants have evolved to absorb light in the visible spectrum, and similarly reflect light with wavelengths greater than $0.7\mu m$. The longer wavelengths have less energy available per photon, absorbing light in these wavelengths would not be beneficial to the plant, and could potentially cause damage to the cells by overheating. The result of the absorption and reflection at different wavelengths means that healthy living plants are bright in the near infrared spectrum and relatively dark in the red spectrum, a property utilised in the popular vegetation index NDVI. The NDVI is a graphical indicator which uses the simple equation below do assign a pixel a value between -1 and 1:

$$NDVI = \frac{(NIR - RED)}{(NIR + RED)}$$

Vegetated areas are expected to give higher values due to the properties discussed, water and snow will yield negative values due to larger reflectance in the visible spectrum than the infrared, and soil should result in values close to zero due to similar reflectivity in both spectrums. Additionally different species of plants have different reflectivity properties across the spectrum, a property which is commonly used to classify vegetation types in remote sensing images (Vohland et al. [2007], Dinuls et al. [2012]).

2.3.1 Classification of Remote Sensing Images

Forest classification from remote sensing images can provide invaluable information for foresty operators, such as species composition, density, crown closure, height, and age (Franklin et al. [2001]).

Automated classification of remote sensing images allows for rapid identification classes present. An example of this is the pixel by pixel classification used of Conifer species composition (Franklin and McDermid [1993]). Another approach is the identification of individual tree crowns to allow quick calculation of figures such as stems per hectare, or crown diameter to assist in this task. For example Pinz Pinz [1991] developed

a method based on the radial brightness distribution of tree crowns. Individual tree counting, while superfluous to the goal of this thesis, is relevant as such methods are able to identify potential trees in aerial imagery from the background.

One particularly relevant approach to tree counting (Yang et al. [2009]) initially performs a pixel-level classification of images into tree and non-tree regions. The researchers note that colour is the most revealing feature for trees, that the texture of trees can be used to distinguish trees from similarly colour objects, and that the entropy of a pixel's neighborhood can be useful for distinguishing between trees and man-made structures. The researchers hence developed a classifier based on colour, texture, and entropy, finding that these features sufficient to correctly identify tree pixels with an accuracy $> 90\%$. Yang et al. [2009] used a number of features in their classifier - 27 features per pixel made up of two colour space transformations, a gaussian filter bank for textures, and three different window sizes for entropy calculations. Calculation of these values would not be feasible on a small computer operating in real time on a video with reasonable resolution. However simpler descriptors of colour and texture could be sufficient for this thesis as the classification task of forest compared or cutover is much simpler than that required in Yang et al. [2009].

Individual tree counting methods are superfluous for this pursuit as individual tree information is not required, and adds unnecessary computational complexity.

2.4 AUTONOMOUS NAVIGATION

Autonomous navigation is a developing field experiencing a large amount of interest recently due to the potential for self-driving cars. Elon Musk, CEO of Tesla Motors has stated that he expects autonomous cars to be a reality within two years Korosec [2015]. Advances in LIDAR technology mentioned in Section 2.1 as well as ongoing development in computer vision tasks such as object detection and identification have enabled this possibility. Several large companies in addition to Tesla are racing to achieve safe driverless control; in January 2015, Google revealed a prototype car without a steering wheel, or conventional pedal controls and outlined plans for testing later in the year. Automotive manufacturers Mercedes-Benz, Nissan, Audi, and Delphi are also in the development stages of driverless cars. Dickmans et al. provide a road detection method using capable of determining nine road and vehicle state parameters at 25 Hz on microprocessors as early as 1992 Dickmanns and Mysliwetz [1992].

It is not just driverless cars that are using a range of sensors and algorithms to allow computer navigation of vehicles, NASA's successful Mars rover 'Curiousity' (Figure 2.10) is another example of autonomous operation of a ground vehicle. In August 27, 2013, the golf cart sized rover successfully used autonomous navigation to drive over new ground which had not previously been declared as safe. The system works by retrieving a set of stereo images which are processed onboard to map any hazards or rough terrain (Webster [2013]).

Figure 2.10 NASA's Curiousity rover on Mars is a famous example of a robot capable of autonomous navigation Webster [2013]

Pilotless aerial vehicles may be further away from full implementation than their ground based counter parts but research into autonomous operation of aerial vehicles is

ongoing. One key challenge UAVs face compared to ground based alternatives is six degrees of freedom (DOF), and operation in a 3D airspace rather than 2D road network. Generally a car only needs to worry about moving two DOF. Fixed winged UAVs have forward/back, pitch, roll, and a degree of yaw. Multirotors UAVs are even less constrained and are able to move in all six DOF — forward/back, left/right, up/down, pitch, roll, and yaw. The additional dimensions increase navigational complexity, for example if a car encounters an obstacle, then it can slow down, swerve left/right. If the same happens on board a UAV then it can slow down, swerve left/right or up/down.

Previous research shows emerging use of machine vision and other techniques for assisting manual flight or for autonomous navigation. Examples include using vision to assist with the low level navigation inertial control of aerial systems (Carrillo et al. [2012]) and assisting with navigation of preset paths (Krajník et al. [2012], Strydom et al. [2014]). A specific example of assisted manual operation of multi-rotor UAVs is work by Saripalli et al. [2002] where video from a down facing camera was used to locate, recognize, and autonomously land a helicopter on a helipad. Thresholding is performed to reduce the computational cost of the algorithm, then a median filter is then applied. Groups of white pixels in the frame are analysed, groups too large and too small are ignored. At this stage there may still be objects which are not the helipad present in the frame. First, second, and third moments of the objects in the image are then used to find the characteristic 'H' of the helipad within the frame — if the object's moments lie within 10% of the stored values for the helipad then the helipad is said to be recognised. The same issue of automated landing was addressed with a slightly different technique by Shakernia et al. [2002] who used a multiple view approach to motion estimation. The multiple view approach uses multiple frames from the same camera to estimate the current position in space. Experimental results Saripalli et al. [2002] and Shakernia et al. [2002] showed successful landing operation of a UAV, Saripalli et al. [2002] was accurate to 0.4 m, with 7° rotation, and Shakernia et al. [2002] was even more accurate to within 0.07 m, and 4° rotation.

A similar approach was used by Frew et al. [2004] for autonomous flight following a road using a fixed wing UAV. Bayesian pixel classification, connect component analysis, Hough transform, and robust line fitting of video from a downward facing camera were used to identify a road. The position of the road (the lateral or crosstrack error) in the frame was then used to determine the commanded turning rate. To control the yaw rate, a nonlinear control strategy was developed based on the difference between the aircraft's heading and the direction of the road. The system was able to correctly identify the road in 90% of images however navigation performance was restricted by lack of attitude information. The pitch and roll of the aircraft were assumed to be zero which limited the performance of the system.

The team further developed their road following work and produced a system capable of following rivers (Rathinam et al. [2007]). The river is detected in either

infrared or colour images using a similar approach as previously. Training is performed offline so that a Bayesian likelihood classifier can be used in real time during flight to classify a river. Connected component analysis is applied to the image to eliminate false positive detected, and then a curve is fitted to the river in the frame based on the region's centers of mass. Flying at a height of 135 m, the team were successful in having their fixed wing craft fly along the a section of a river, with an average crosstrack error of 7 m. Figure 2.11 shows a graphical representation of the process; this image is interesting as it shows a work flow which would be suitable for processing forestry images to find the cutover edge for navigation.

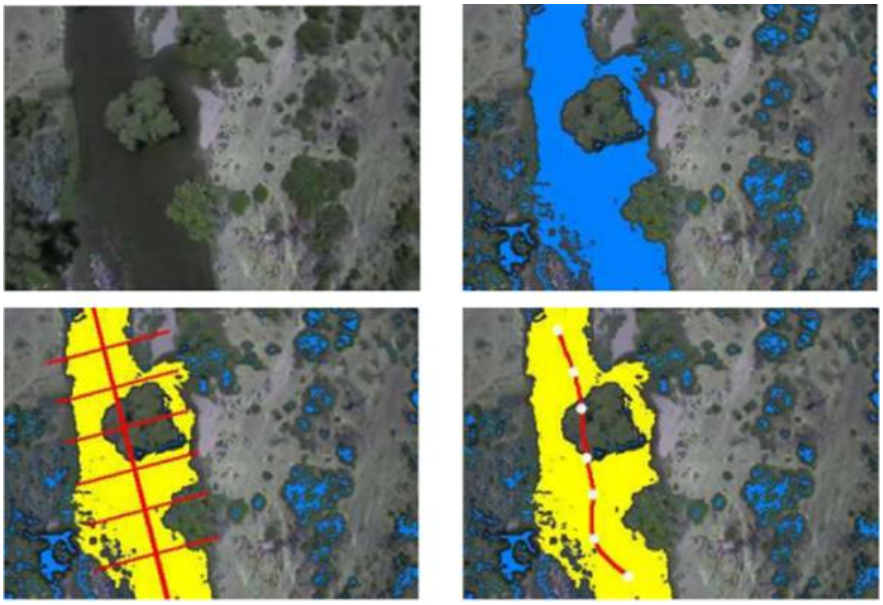


Figure 2.11 An image from Rathinam et al. [2007] showing the process of extracting navigation information from a target image. The blue areas show the pixel classification result, yellow shows the identified river, and the line shows the navigation curve based on the region's center of mass.

Baker and Kamgar-Parsi [2010] achieved autonomous flight using a fixed wing UAV for following coastlines using a near infrared camera to detect water. In the NIR images, a histogram of all pixel values is generated and used to determine the likely maximum value of water has in the image. A binary classification system of water or non-water is then used, areas with little variation and values less than the water's maximum value are marked as water. Due to the use of a fixed wing UAV, the guidance system used turn maps in order to navigate the length of the coastline. The turn map allows the system to find a rate of turn so that it will intercept the shore line on a course tangent to the shore line. The system has knowledge of it minimum turn radius so when it sees a sharp turn like a spit of land, turning will begin earlier rather than following the spit and getting lost when unable to turn back in time. The navigation system did however struggle at maintaining positioning above the coastline and would need further development in order to be used the task at hand.

Chapter 3

PROPOSED DETECTION AND NAVIGATION ALGORITHMS

3.1 ACQUIRING CUTOVER IMAGERY

3.1.1 Aerial Photography from Manned Survey Aircraft

In order to assess the most appropriate type of imaging to pursue it was important to gather aerial imagery of the forestry cutover. Collecting images of each spectrum simultaneously allowed direct comparison of images without factors such as lighting conditions and forest properties influencing the results. A small Piper Cub plane with an aerial photography rig was used for collecting images. A modified Canon 20D and a multi spectral Tetracam ADC were fitted in an aerial photography mount that hangs from the side window of the plane, as seen in Figure 3.1.

In order to capture images in a stereovision setup, two Canon 100D cameras were mounted on the wing struts of the piper cub, giving an interaxial separation of 4 metres. The large interaxial separation was influenced by the very limited mounting locations on the underside of the plane and the resolution requirements of identifying a 30m altitude difference between the forest and the cutover. Stereovision requires images to be captured at precisely the same time and so a small battery powered camera triggering system was built using a PICAXE microcontroller. The system was set to trigger three outputs simultaneously at two second intervals once activated. A video feed from one

(a) Canon 100D mounted to a wing strut. (b) Mounting plate for aerial photography.

Figure 3.1 Camera setup for the aerial photography flight.

Figure 3.2 The triggering system being tested prior to the aerial photography flight

(a) View of 'Sweetie Pie' the Piper Cub as she is rolled out for the flight.

(b) View from the passenger seat of the right wing strut and camera mount during flight.

Figure 3.3 Images of the survey flight

of the cameras was used to monitor the position of the forest underneath the plane to ensure that cutover edge was captured. The time interval had been calculated to give an overlapping section between sequential images, while still allowing time for the image capture and writing processes. Unfortunately the height above ground was lower than anticipated and so only a few images contained any overlapping regions, consequently it was not possible to use image stitching software to produce a composite image mapping a large area of the cutover edge. Another setback of the flight was that one of the Canon cameras in the stereo arrangement which had been problematic prior to the flight suffered an error in the first few seconds of the test flight, capturing only five images of the ground. Figure 3.2 shows the triggering system being tested.

The aerial photography test flight took place on the 16th of April at approximately 1-2pm. There was no cloud cover and minimal wind. Survey of two regions was performed, Eyrewell forest, and Loburn Forest. Eyrewell forest is located just north of the Waimakiriri River in the North Canterbury plains. Due to ongoing operations to clear the forest to replace with dairy farms, the forest edge had suitable cutover edges with the cutover side cleared of material. There was also an example of grass growing on the cut edge shown in Figure 3.4 which while not common in forestry operations (Graham [2015]) proved to be useful in the development of the edge detection algorithms. Unfortunately, all the cutover edges at Eyrewell are straight, as is visible in Figure 3.4,

limiting the scope of potential software testing using the raw imagery.

Figure 3.4 An example image from the flight over Eyrewell Forest.

Loburn forest is located in the hills inland from Eyrewell forest. Unlike at Eyrewell there were not very clear cutover areas, and most forest edges had significant vegetation growth as is visible in Figure 3.5a . Imagery collected at Loburn Forest was useful for testing the boundaries of the developed forest detection algorithms. The hilly nature of the landscape made maintaining a constant height above ground impossible, which was fortuitous as it meant images were captured at a range of heights above ground. The forest at Loburn Forest was particularly irregular and contained pine plantations of different levels of maturity, shown in Figure 3.5b.

When comparing footage from the RGB, infrared, multi-spectral cameras, it was found that images captured in the RGB spectrum provided the greatest distinction between the forest and the ground on the cutover. Extra costs and complexity associated with working with the much less common infrared or multi spectral images were hence avoided. Due to the failure of one of the canon cameras, the efficacy of stereo vision was not determined on this survey trip.

(a) Image showing shrubbery with pine forest.

(b) Image showing contrast in tree age present.

Figure 3.5 Images captured of Loburn forest.

Figure 3.6 RGB image of cutover.

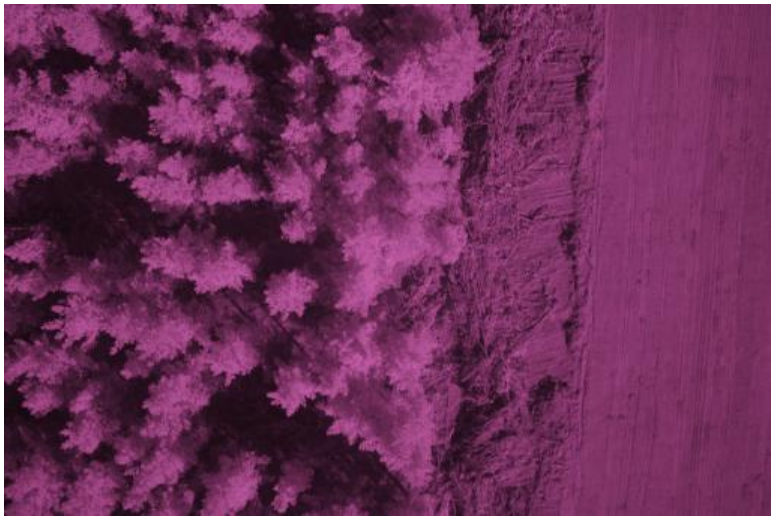


Figure 3.7 Infrared image of cutover.

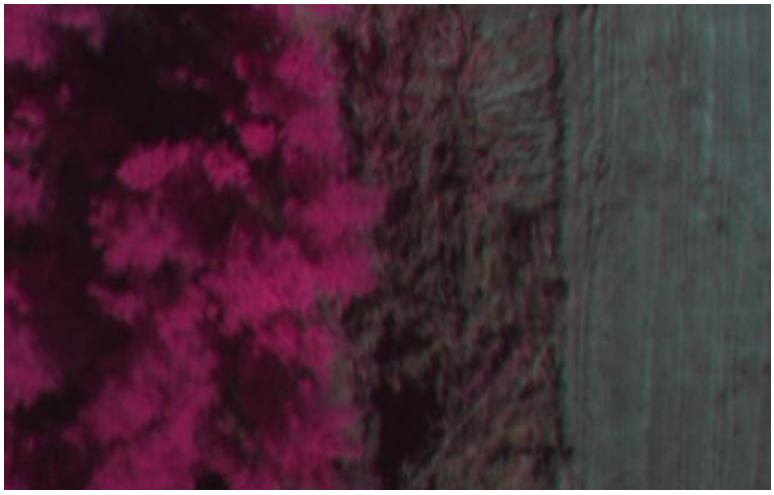


Figure 3.8 Multispectral image of cutover

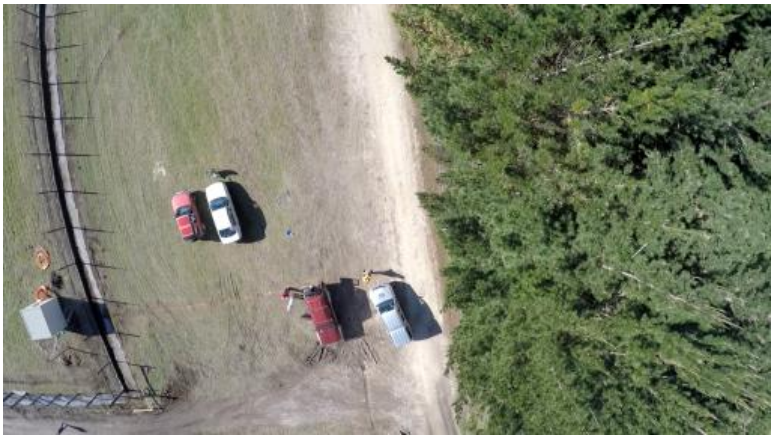


Figure 3.9 Frame grab from UAV footage of forest edge at approximately 50 m, provided by SCION.

3.1.2 Aerial Photography from UAV

SCION assisted in the gathering of imagery for this project by providing video footage from a UAV of two different forest edges. The videos were particularly useful when testing out the performance of developed algorithms as the footage was of a different cutover edge, camera, and altitude to the images used to train the support vector machine classifier (Section 3.2.2). Both videos were recorded using a GoPro camera recording at 1080P resolution and 24 frames per second (FPS). Based on the scale of features in the video compared to the second video, the height above ground of the first flight was estimated to be 50m.

The second video was provided by SCION at a different location. The same edge was flown three times, following the same flight path at 60 m , 80 m, and 100 m. Section 3.1.2 shows a frame from the video of a similar position at each of there altitudes. Barrel distortion characteristic of GoPro's wide angle lenses is apparent in both videos provided.

3.2 DEVELOPMENT

A fundamental part of this project was the development of algorithms which are able to correctly identify a forest edge and use that information to make sensible navigation commands for sending to the flight computer for control of the UAV. The software required to perform this task can be separated into three categories: forest and edge detection, navigation, and communication.

Forest and edge detection involves the identification of forest edges using one or more methods and is required to be robust to varied lighting and forest conditions, and cutover properties. Navigation involves using the previously identified edge location to produce outputs describing the required movement of the UAV. The navigation algorithms are required to be versatile to a number of different edge conditions, and must

(a) 60 m

(b) 80 m

(c) 100 m

Figure 3.10 Frame grabs from UAV footage of cutover edge at three different heights above ground, provided by SCION.

produce sensible outputs as they are used to directly control a UAV without operator input. The communications section encompasses the link from the onboard computer to the Pixhawk flight computer. Open source software produced by Pixhawk developers was adapted for this section, allowing for the project to focus on the development of proposed forest edge detection algorithms and navigation sections.

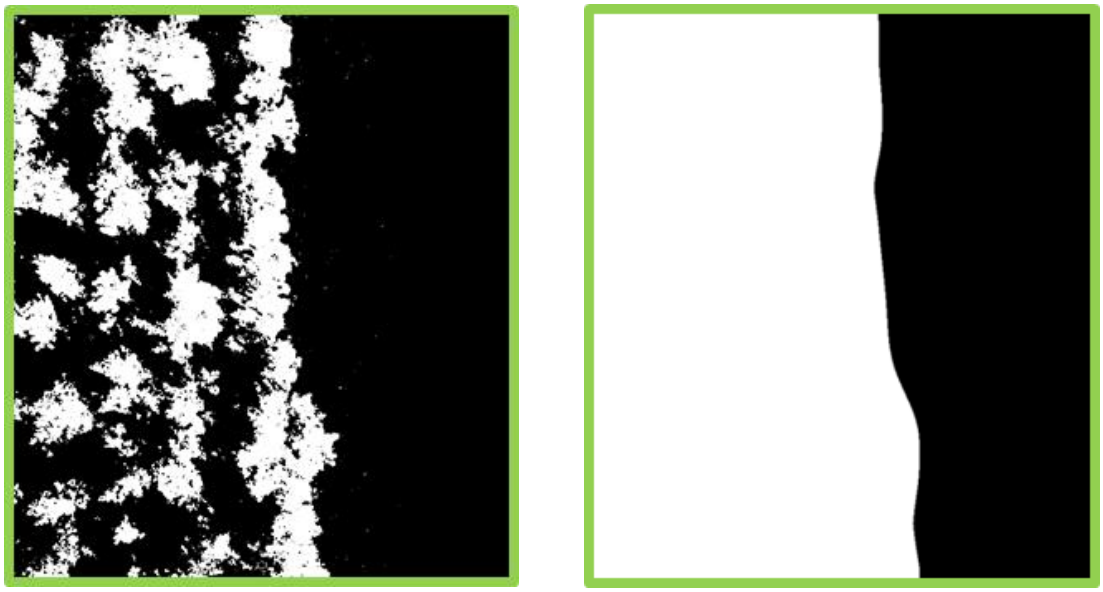
3.2.1 Proposed HSV Forest Detection Method

Several different methods for the detection of forest in an image or video were explored. The performance of each was reviewed to ascertain the most appropriate for use on the onboard computer. Aspects considered were reliability/robustness, speed, and the resolution/quality of the edge data. The first method developed for detection of forest was a selection of a colour range using thresholding and additional filtering. The process of the colour based forest selection is as follows:

1. Load RGB image from camera,
2. Convert raw RGB image to the hue-saturation-value (HSV) colour space,
3. Select forest by binary filter of a particular range of HSV values (found by experimentation),
4. Reduce noise of binary image by median filter, morphological operations, or some combination of both,
5. Detect contours in the image and select only the largest closed contour.

The colour space conversion from RGB to HSV allows for the intuitive selection of a colour based on hue, saturation, and value (also referred to as the brightness). A hue range of 35–75, a saturation range of 30–150, and a value range of 10–190) were found to be effective. Any pixel in this range is changed to white, while all others are changed to black. The noise reduction filtering is required to fill in the gaps between white areas caused by shadows between trees, as seen in Figure 3.11.

The colour based forest selection method gave high quality edges and due to its simplicity it is fast enough to operate in real time. The main problem with the method is that it is very susceptible to false positives when other green objects are in the frame. Examples of this are when there is regrowth on the cut edge such as small bushes or grass. Step five was added to reduce the effect of small green patches on the cutover by comparing their size to the forest in the frame. Selecting the biggest contour was not a completely robust solution however as the classifier can still break down in certain situations. If there are areas of green very close to the edge, then the position of the edge will be warped. Additionally, if a non-forest region of green on the cutover is larger



(a) Before filtering. (b) After filtering.

Figure 3.11 Forest masks generated using colour selection method.

within the image frame than the forest, then the non-forest region (or contour) will be classified as forest.

Discussing the issue of colour selection on the cutover imagery with SCION revealed that generally there is not vegetation present on the cutover as forestry operations proceed fast enough that there is minimal regrowth. However, due to the potential for research performed on this project to be used in industry, as opposed to a purely academic pursuit, methods developed require robustness and versatility so it was decided to continue investigating other methods.

3.2.2 Proposed Support Vector Machine Classifier Method

Literature review revealed that a priori information can be used to train a classifier utilising texture descriptors in addition to colour information when performing image classification. A support vector machine (SVM) was trained using the mean and standard deviation of each colour channel, a total of six values per training image. The training images used were 48 pixels $\times$ 48 pixel, a size chosen as it is a multiple of two and so can be used with a large number of step sizes, and as a balance between size and resolution. Larger images would be result in fewer images and hence faster processing, but would not allow for the same resolution which could be detrimental to the quality of classification. For example a large image containing both forest and cutover would likely be classified as not forest due to the collection of average features, resulting in the detected edge being out of position compared the actual edge. It was originally planned to test some different image sizes however the 48 pixel square proved effective so this

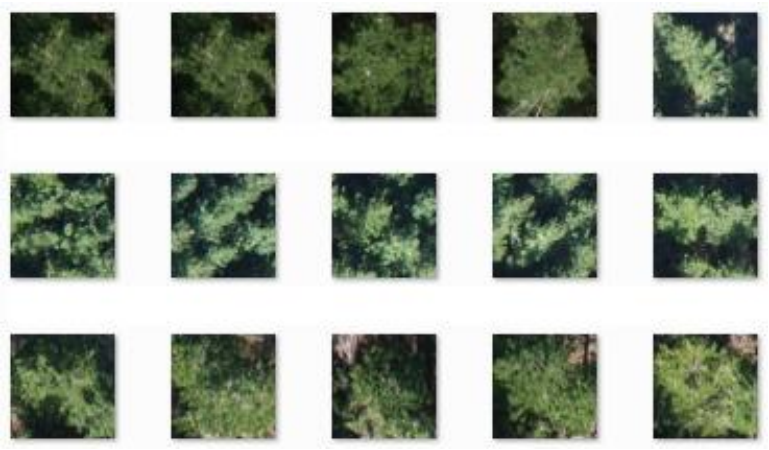


Figure 3.12 A selection of positive training images including forest

was not performed. Positive images were scaled and cropped from some of the BPK flight imagery. A selection of the positive training images are shown in Figure 3.12. Negative images were taken from the BPK flight and also from miscellaneous Internet images. Images from the flight of shrubbery and grass were included in the negative training set to reduce the number of false positives. 80 positive and 80 negative images were collected in total. The mean and standard deviation for each colour channel were calculated and put in a vector, giving a length six vector for each image. Training was then completed following the guidelines set out in Hsu et al. [2003].

The process for forest classification using the SVM classifier is as follows:

1. Load RGB image from camera, and create a blank forest mask image.
2. Calculate the mean, and standard deviation for each colour channel for a 48 pixel $\times$ 48 pixel window.
3. Query if the window is forest using the classifier, if the window is positive then lighten the corresponding area on the forest mask image, if negative then darken the corresponding area.
4. Slide the window by the step size (less than or equal to the image size), and repeat step three. Repeat until the entire image has been processed.
5. Threshold the binary forest mask image to get a black and white image.

By using a step size smaller than the window size, each pixel within the image is classified multiple times rather than just once as would occur if the step size were equal to the window size. Each pixel value hence lies on a spectrum from 0-255, with higher values (whiter) indicating higher confidence in classification as forest. A benefit of this approach is it allows for thresholding at a given level of confidence, reducing the effect

of false responses. Unfortunately, as the step size decreases, the processing time per frame increases by a power of two, as seen in Figure 3.13.

The initial results of the proposed SVM classification method demonstrated a clear improvement in forest classification. Increased performance was due to the SVM method's ability to discard regions of green on the cutover based on texture. The SVM classifier occasionally gave false positives on unexpected objects, such as vehicles on the cutover edge. It was also noticed that the classifier breaks down at the edges of images in areas of high distortion due to the wide angle of the Gopro used by Scion in gathering of sample imagery. Radial and linear kernels were both trialed. The differences between kernels were minor; the linear kernel appeared to have a slightly lower false positive classification rate. A comparison between the two kernels has been shown in Figure 3.14. It can be seen in Figure 3.14b that the radial filter leaves a few dark patches in the middle of the forest area, while the linear filter in Figure 3.14a shows fewer darker patches.

The performance of the SVM classifier using the mean and standard deviation of each colour band was such that additional texture descriptors were not introduced. While there is definite scope for exploring alternative or additional descriptors in the future, this was not explored as the mean and standard deviation provided sufficient classification accuracy and importantly are quick to calculate. The affect of additional descriptors on processing time should be considered in any future development on texture descriptors.

3.2.3 Optical Flow

Both the colour and SVM classification methods rely purely on static image data. In order to infer additional information about the environment below, one idea was to use the motion of different pixels or features between frames of video footage. It was hoped that this would facilitate the identification of the edge, based on the sudden height change between forest and the cutover edge. One would expect feature points or pixels closer to the camera i.e. the forest canopy, would generate a greater disparity between images compared with features or pixels further from the camera i.e. cutover surface, as the closer features would move through the frame faster.

Using multiple images to infer 3D structure is the basis of structure from motion (Section 2.1.2.2). However due to SfM's associated computational complexity and the limited processing power available mean that it is not well suited for this situation. With preference for real time operation, an optical flow technique was tested in preference to SfM. The optical flow algorithm described by Farnebäck [2003] was used to test the feasibility of using optical flow classification. The outcomes showed that the method was not very effective, primarily because any discernible difference between motion of pixels on the cutover and forest canopy was incomprehensible during strong pitch, roll, or yaw

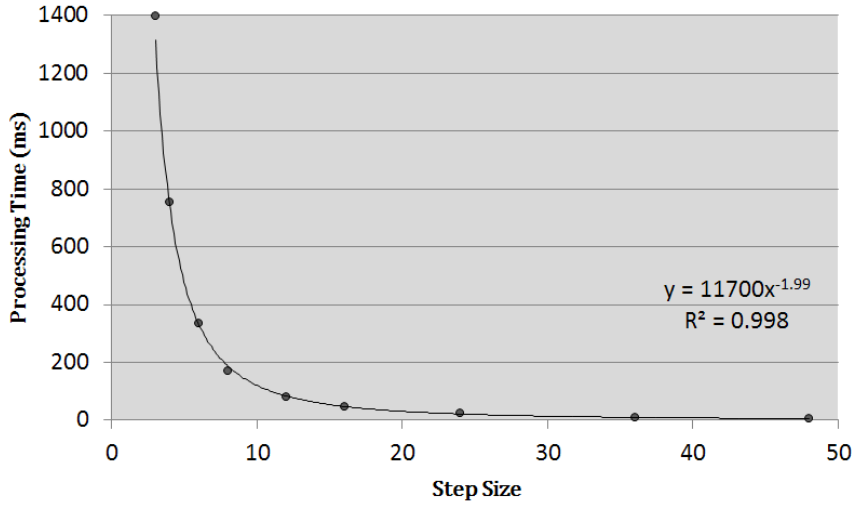


Figure 3.13 Processing time for SVM forest detection on 672×378 image.

(a) Linear kernel (b) Radial kernel

Figure 3.14 SVM forest classification with areas testing negative darkened, and positives areas unaltered.

of the camera. Figure 3.15 shows the disturbed output during roll and yaw movements respectively. Unfortunately, the requirement of generating navigational instructions 'on the fly', so to speak, in a closed loop situation means that there is no assurance of camera stability and hence image quality. Due to the described flaws encountered with this method, optical flow results were not produced for comparison with other methods.

An interesting avenue for exploration in future work would be simultaneous localisation and mapping using SfM (as described in (Mouragnon et al. [2009]) in an aerial environment for navigational assistance. Continuous advances in software algorithms for SfM mean it has become plausible for such a system to run on a small UAV with limited processing power, however due to time restrictions this has not been explored further.

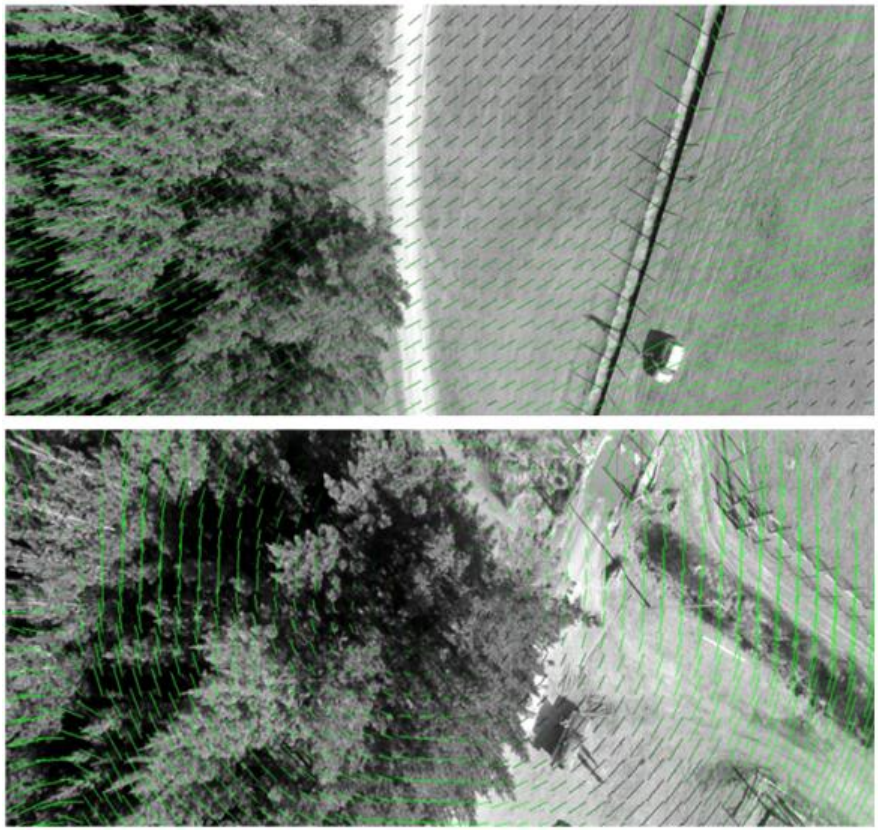


Figure 3.15 Farneback optical flow output during roll and yaw disturbance respectively. The lines represent the movement of pixels between frames, with brighter lines indicating greater movement.

3.3 PROPOSED CUTOVER EDGE IDENTIFICATION ALGORITHM

A combination of the proposed colour selection and SVM forest classification algorithms was used to accurately detect the cutover edge in aerial imagery onboard a UAV. The SVM classifier provided additional accuracy when dealing with regions of similar colour

to the forest, however the SVM classification results did not capture the fine details of the cutover edge as well as simpler colour based classification.

3.3.1 Edge Identification

The colour detection and SVM classification methods are both used to select sections of an image thought to include forest. The information classification outputs are stored in forest mask images for each classifier. The forest masks are binary images with white representing a positive classification of the pixel as forest. In order for this information to be useful for navigation, the edge of each section of forest needs to be identified in a timely manner. This was initially performed by iterating through columns in each row until there was a change from white (forest) to black (not-forest), or vice-versa. If there was more than one change, then that particular row was ignored. Rows with more than one change in forest were ignored as it would be difficult to know which changing point in that row corresponded with the edge of the forest. This was reasonable method for identifying the edge in most cases when classification accuracy was high. However, when there were large areas in the forest that were not correctly identified, the line could be skewed significantly.

A refinement of the edge identification method was developed in order to improve the edge identification by combination of both classifiers. The refined method is as follows:

1. Detect forest using SVM classification method,
2. Detect forest using colour selection method,
3. Using SVM classification forest mask, determine whether the forest lies on the left or right hand side of the image by counting number of white pixels. (This is only required once as the nature of the cutover edge means that the forest will not change sides within the image unless the UAV follows the edge in the opposite direction),
4. Find change from black (not-forest) to forest, searching from the not-forest side to forest side using colour selection image,
5. Fit a line to the SVM and colour selection forest masks, then combine.

Identifying which side the majority of the forest lies on is useful as it is used to determine whether we should scan in the left-right direction or the right-left direction. It was observed after running both algorithms on the cutover edge UAV video that areas of shadow between trees caused more false negative classifications than there were false positive classifications. By changing scan direction the trend of fewer false positives is exploited in order to improve the edge quality. An example of this is visible in **??**,

(a) Colour detection forest mask.

(b) SVM detection forest mask.

(c) Cutover edge identification output showing colour selection(Blue), SVM classifier (Green), and combined linear output (Red).

Figure 3.16 A cutover edge detection example showing forest detection masks for both colour selection and SVM classifier, and final identified edge position. Note that the presence of a dead tree in the cutover has not impaired the algorithm's edge detection quality.

which shows edge detection output and corresponding SVM and colour selection forest masks. Both raw forest masks contain an unclassified black section on the cutover due to a dead tree. The forest is found to be on the right hand side of the image using the SVM forest mask, and so both masks are searched from left-right to find the edge. The small change to the calculation of edge position adds negligible computation time and improves edge detection robustness.

Simple linear regression was used to fit a straight line to the identified edge. Fitting a trend line to the edge data previously generated reduces the effect of outliers in the edge data. A linear trend was used due to its simplicity and speed of calculating; it was decided that higher order trend lines would add additional computational complexity, and would be more difficult to interpret, without adding any significant benefit. The linear regression was performed on both the colour selection and SVM methods. This allowed for the calculation of the weighted mix of both classifiers for the final output – smoothing the affect of a performance drop in either classifier. The linear trend line of the last x frames is averaged in order to further smooth the output to the flight controller. A rolling buffer was implemented to perform calculation of the mean trend line.

3.3.2 Proposed Navigation Algorithm

Development of a navigation algorithm was required to translate the output of edge fitting into a velocity or position command interpretable by the Pixhawk flight computer. The intersection of the fitted linear trend line with a horizontal line through the centre of the image was used to determine the positional error. The gradient of the fitted trend line was used to determine the required yaw correction to maintain forward flight over the cutover edge. The navigation algorithm uses proportional control - the output is based on the number of pixels away from the centre the line is multiplied by some gain value, meaning that the UAV will move faster towards the detected cutover edge if it is further away. More advanced control mechanisms such as differential and integral control could be investigated further however it is not anticipated that they will provide significant benefit due to the course, constantly changing nature of the cutover edge.

The forward velocity is held at zero until the cutover edge has been successfully identified, following edge identification the UAV accelerates up to a predetermined maximum velocity, during final testing on this project the maximum velocity was set to 3 ms^{-1}. The forward velocity is reduced if the edge quality falls below a predetermined threshold, giving a larger number of processed frames for a given section of forest.

As the edge detection algorithm is running at multiple times per second, techniques were applied to smooth the output to the flight computer:

- Positional and rotational dead zones were implemented to reduce oscillation about the cutover edge,
- Maximum output values were introduced to facilitate smooth flight,
- The moving average of multiple previous frames discussed previously aims to reduce the effect of any sudden change in the output of the edge detection algorithm,
- Different flight states are used to manage different situations and their velocity output requirements, e.g. If the edge detection is unable to identify the forest edge for 10 sequential frames then the UAV enters the “EDGE LOST” flight state, and begins slowing down while maintaining current heading. This gives the edge detection opportunity to re-find the edge, while reducing the disturbances to the camera input from any sudden deceleration.

In summary, the navigation method proceeds in the order shown:

1. Check if edge has been found, if not then set outputs to zero.
2. Convert gradient to angle.
3. Multiply positional and angular error by their respective gains.

4. Check if edge position or rotation output are in their respective deadzones.
5. Convert to NED reference frame.
6. Check if position or rotation output is above maximum.
7. Send Mavlink navigation command to the Pixhawk flight computer.

3.3.3 Conversion of Navigation Commands into Required Reference Frame

Pixhawk (see Section 4.1.4) accepts offboard instructions in the form of Mavlink commands. The autopilot interface code (Meier and Lukaczyk [2012]) created by the Pixhawk development community was used to deliver velocity commands using the 'SET POSITION TARGET LOCAL NED' command. The required inputs into the send velocity function are velocities in the North-East-Down (NED) coordinate frame. As the camera is facing down at an attitude determined by that of the quadcopter, conversion from left/right to North/East is required. In order to transform the navigation instruction into the NED coordinate frame, the heading is requested from the Pixhawk, then the following transform is performed on the onboard computer:

$$globalEast = cos(heading) \times localX + sin(heading) \times LocalY$$
$$globalNorth = -sin(heading) \times localY + cos(heading) \times LocalY$$

Chapter 4

HARDWARE, AND IMPLEMENTATION

4.1 HARDWARE SELECTION

4.1.1 Laser Rangefinder

The system included a laser range finder for maintaining constant height above ground to ensure safe operation of the UAV. The laser range finder selected was a Lightware SF10-C, shown in Figure 4.1. The key features of the device are:

- Specifically design for use on UAVs,
- Range of up to 100 m,
- Low mass (35 g),
- Class 1M laser, considered non-hazardous unless viewed with optical aids,
- Multiple connectivity options (analog, serial and i^2c).

Figure 4.1 SF 10-C laser range finder.

The SF 10-C range finder requires very little effort to integrate into the system. In order to activate the range finder, the line "sf0x start" is added to the "extras.txt" file

on the pixhawk flight controller. While software developed during this project had the capability of using the laser range finder to set the UAV's altitude, it was not tested as software testing was a on cutover edge. Use of laser range finders for maintaining a constant height above ground has been discussed in .

4.1.2 Onboard Computer

The computer onboard the UAV was required to run forest edge detection algorithms which, due to their high computational cost are not able to be run on the Pixhawk flight computer. Several options including the Minnowboard Max were considered, including the Raspberry Pi 2, the BeagleBone Black, Intel Nuc, Odroid. The Minnowboard Max was selected based on its balance of processing power and size. The minnow board runs a 64 bit Intel E3825 dual-core processor, 2 GB ram, while only weighing 63 grams. The number of available signals including I2C, GPIO, and PWM mean that the Minnowboard is suitable for connecting with a flight computer and other accessories such as a laser range finder. Following the crash of the system during flight testing, the Minnowboard Max suffered an impact to the CPU heatsink. Fortunately as the system runs from an SD card onboard data was recoverable despite irreparable damage to the board. It is recommended that recoverability of data in the case of a crash is considered in if an alternative board is used in future work. The Minnowboard Max was replaced with the later mode - the Minnowboard Turbot. The Turbot has an identical layout and very similar features but has the more recent Intel E3826 processor and a few other minor improvements.

Figure 4.2 Minnowboard Max single board computer.

4.1.3 Video Transmitter

The video transmitter is a highly useful tool for assessing cutover edge detection performance and UAV positioning during flight. Both Minnowboard models have a micro HDMI video output so the selected transmitter required an HDMI input. A match was found in the Skyzone TX-5D HDMI to AV Transmitter. The transmitter takes an HDMI mini input and converts the digital signal to analog before transmission to a ground station. A micro to mini HDMI cable was purchased to convert from the Minnowboard output to the Skyzone transmitter input. Figure 4.3 shows an image of the transmitter with the HDMI cable.

Figure 4.3 Skyzone TX-5D transmitter used for flight testing of the system.

4.1.4 Flight Computer

A pixhawk flight computer was chosen for use on this progress as it is an open source, open hardware, autopilot-on-module that is commonly used by both for research, industry, and hobbyists.

The Pixhawk hardware can either run the PX4 or APM flight stacks. It was decided that the PX4 flight stack would be used as there was more documentation available on interfacing with the system for offboard control using an external computer. Developer features such as offboard control can be enabled by flashing the developer or master versions of the flight stack. Although the APM flight stack appears to be more fully

Figure 4.4 Pixhawk flight computer.

developed for regular fliers, it does appear to have limited capability for more advanced features.

4.1.5 UAV Used

The Spatial Engineering Research Center at the University of Canterbury who supported this project have a Steadidrone QU4D (steadidrone.com [2015]) which was used for this project. The SteadiDrone QU4D (steadidrone.com [2015]) is a moderately sized quadcopter (diagonal width of 645 mm) with a maximum payload of 800 g. Modifications were required to fasten the afore mentioned hardware as the quadcopter was previously running an older flight controller.

Figure 4.5 The Steadidrone QU4D Quadcopter which was used for all flight testing.

4.1.6 Camera and Gimbal

A small Logitech Quickcam Pro 9000 webcam was used for test flights. The camera was chosen as due to its low mass, low cost, and reasonable video resolution – features making ideal for a proof of concept platform. In the future work it is anticipated that a higher grade machine vision camera such as the PointGrey Firefly will be used but for now the Logitech webcam is suitable for a proof of concept platform. The technical specifications of the camera are:

- Size: 90 mm × 32 mm × 40 mm
- Weight: 147 g
- Maximum video resolution: 1600 × 1200 (4:3)
- Maximum frame rate: 15 frames per second (hardware limit)
- Field of View: 75°
- Focal length: 2 mm

(a) Unaltered Logitech Quickcam Pro 9000

(b) Stripped Logitech Quickcam Pro 9000 mounted to gimbal.

Figure 4.6 Camera used for UAV flight testing.

The camera was stripped down to save weight and to facilitate mounting to a 2-axis gimbal. A 2-axis gimbal is one where the pitch and roll may be controlled or, as in this case, stabilised with the ground plane. Use of a gimbal increases the image quality and usability as there is minimal disturbance to the camera when the quadcopter pitches or rolls. The gimbal has rubber vibration isolating mounts to reduce the transfer of vibration from the motors through the frame to the camera. A 2-axis rather than a 3-axis gimbal was selected (Figure 4.7), as a 3-axis gimbal stabilises yaw, an undesirable characteristic for this application. The act of stabilising yaw decouples the rotation of quadcopter and the camera when viewed from above; the navigation algorithms assume that the rotation when seen from above is the same for both the camera and quadcopter as the rotation of the scene as observed by the camera is what determines the rotational correction sent to the flight computer.

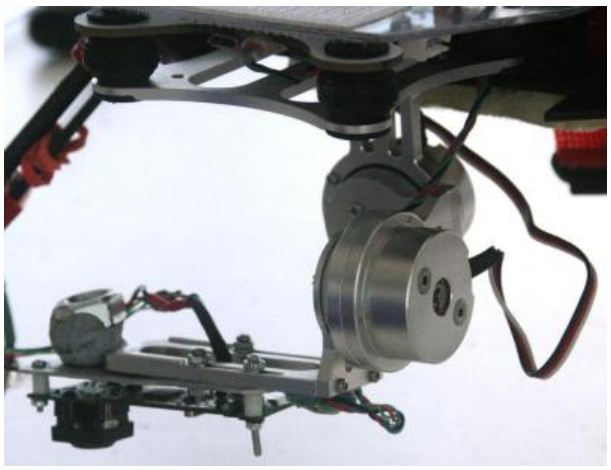


Figure 4.7 The two axis gimbal used.

4.2 SYSTEM DESIGN

It is crucial that each aspect of the system works well with the system as a whole. The simplified system design has is demonstrated in a diagram, shown in Figure 4.8. The Minnowboard Max main loop is primarily made up of the example code from a Mavlink interface example (Meier and Lukaczyk [2012]), with minor modifications made to ensure compatibility with the image processing module. One aspect changed was the offboard mode activation behaviour. The mavlink interface communicates with the Pixhawk and by default immediately activates offboard control mode. This was altered so only the pilot has the ability to change the control mode to offboard mode.

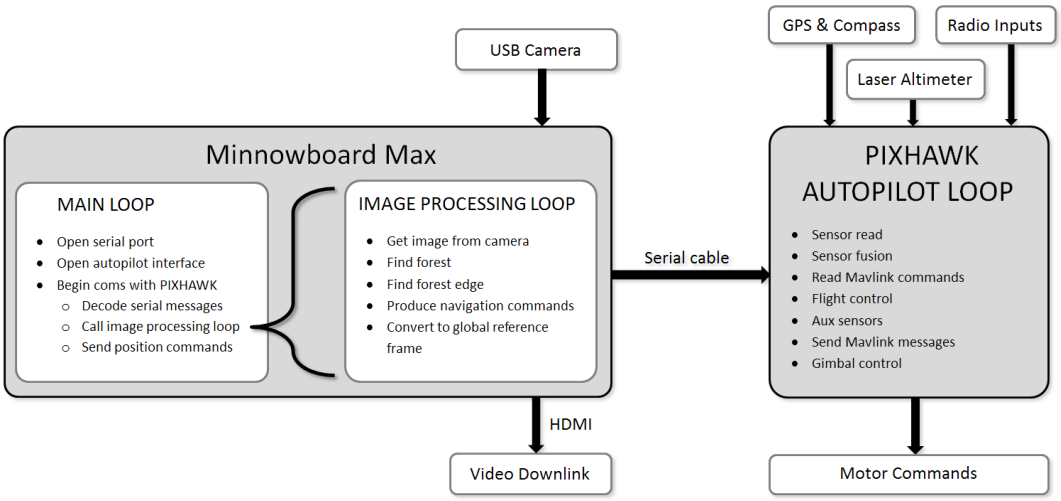


Figure 4.8 The proposed system design showing the integration of the cutover edge detection with the other hardware and software features.

Chapter 5

TESTING AND RESULTS

As described in the development section, the software required for this project can be separated into three categories, forest detection, navigation, and communications:

- Forest detection involves the identification of forest edges using one or more methods and is required to be robust to different lighting conditions, forest conditions, and cutover properties.
- Navigation involves using the previously identified edge location to produce a direction and velocity output. The navigation algorithms are required to be versatile to a number of different edge conditions, and must produce sensible outputs as they are used to directly control a UAV, without operator input.
- The communications section encompasses the link from the onboard computer to the Pixhawk flight computer. Open source example software was used extensively for this section, with minor tweaks made as required; allowing focus on the development of forest detection and navigation sections.

Software from each category, and of the interaction between them was tested prior to any test flights with an actual UAV to ensure the safety and effectiveness of methods developed. The software testing was invaluable for the development of the forest detection and navigation algorithms, and the hardware in the loop (HITL) testing was useful for configuring the correct software and hardware parameters for communication between the Pixhawk flight computer, the onboard computer, and the user.

5.1 FOREST CLASSIFICATION TESTING

The performance of the proposed colour based forest classifier, and the proposed SVM forest classifier were measured by comparison with a series of five manually classified binary images as ground truth. Manual classification was performed in image editing software to show areas of forest in white, and non-forested areas in black. The test

images were chosen to include different brightness levels, shadow direction, cut edge clarity, and height above ground. Images one, two and four have a straight edge; images 2, 3, 5 have shadows across the edge; and images; 2, 4, 5 have green material on the cut edge. Image four was taken was taken from a video recorded from a UAV flight performed by Scion, Rotorua. Due to the similarity of imagery from Scion's UAV flight, only one frame from the video was used with the remaining images sourced from the aerial photography flight. The test images used are shown in Figure 5.1.

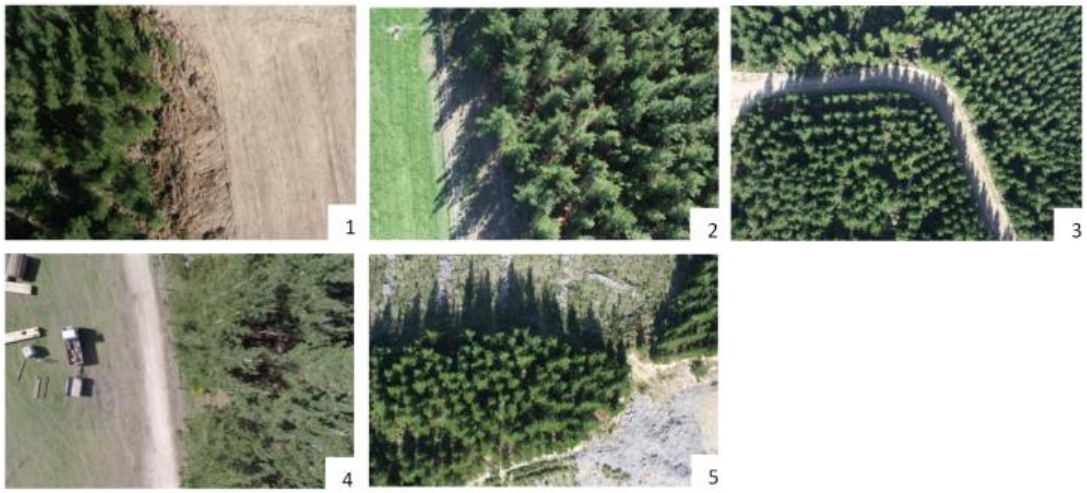


Figure 5.1 Images used for testing forest detection methods.

The colour selection and SVM methods were applied to each raw image. The ground truth images produced were meant as an approximate guide to help assess the performance of the each classifier. A XOR operation on the images was then used to calculate the number of pixels difference between the ground truth and each method's output. A diagram showing this process is shown in Figure 5.2.

The results of the classification performance test presented in Table 5.1 provide an interesting insight into the performance of the two classifiers. In simple images such as images one and two, the colour classifier narrowly outperforms the SVM classifier. However the colour method only outperforms the SVM method by a large amount in one of the tested images, and has a mean accuracy rate that is 7% lower than the SVM classifier. The SVM classifiers poor performance on image four is likely due to the distortion of the trees due to the low altitude of the image; images one and two are more typical of the scale used in the training data set. Classification accuracy of image three using the colour method was particularly low due to the two regions of forest in the image. Due to the contour selection step of the colour classification, the entire forest section below the road is ignored. This highlights the need for a classifier that does not rely solely on colour values.

It is difficult to compare these value with prior literature due to the low altitude of the imagery, and binary nature of the classification. In order to give context to the

Table 5.1 Accuracy of SVM and colour based forest classifiers on five ground truth images.

Image Number	SVM Accuracy	Colour Accuracy
1	96.2%	97.1%
2	94.6%	96.9%
3	92.8%	55.0%
4	81.9%	90.2%
5	85.3%	76.7%
Mean	90.2 %	83.2 %

levels of accuracy attained, a comparison has been made to several Pal and Mather [2005] as researchers were using a SVM classifier (among others) to classify remote sensing data. Take note that the work differs as it used multispectral data from Landsat 7 and was classifying forest in addition to a number of other classes. The accuracy levels attained in Pal and Mather [2005] using multispectral data were 87.9% for the SVM, and 85.1% and 82.9% for an artificial neural network (ANN) and maximum liklihood (ML) classifier respectively. Both the SVM and ANN achieve higher accuracies than the proposed colour based method, but do not reach the accuracy of 90.2% achieved by the proposed SVM classifier. While a direct comparison to prior literature may not be

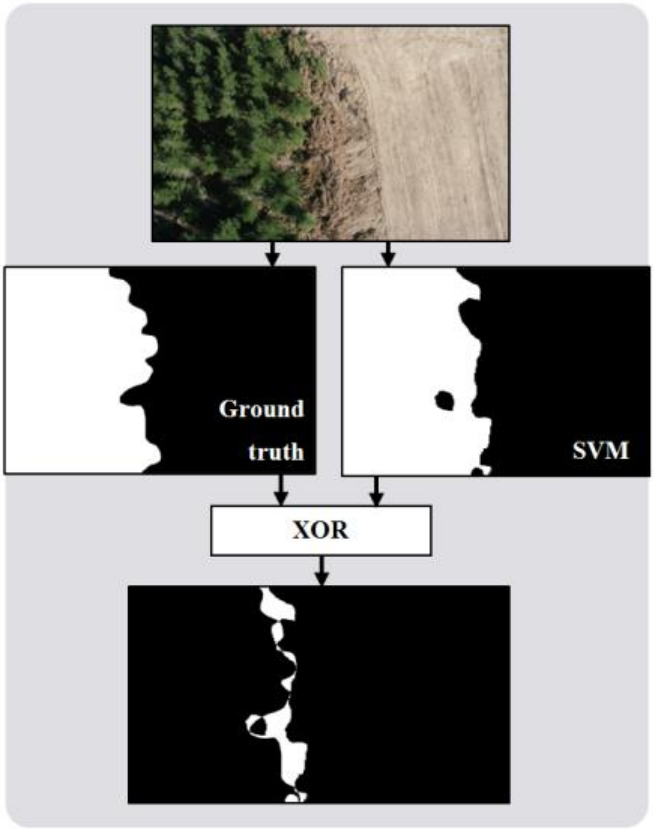


Figure 5.2 The process of assessing classification performance.

valid, the approximate values are indicative of effective classification performance of the proposed algorithms, particularly of the SVM method.

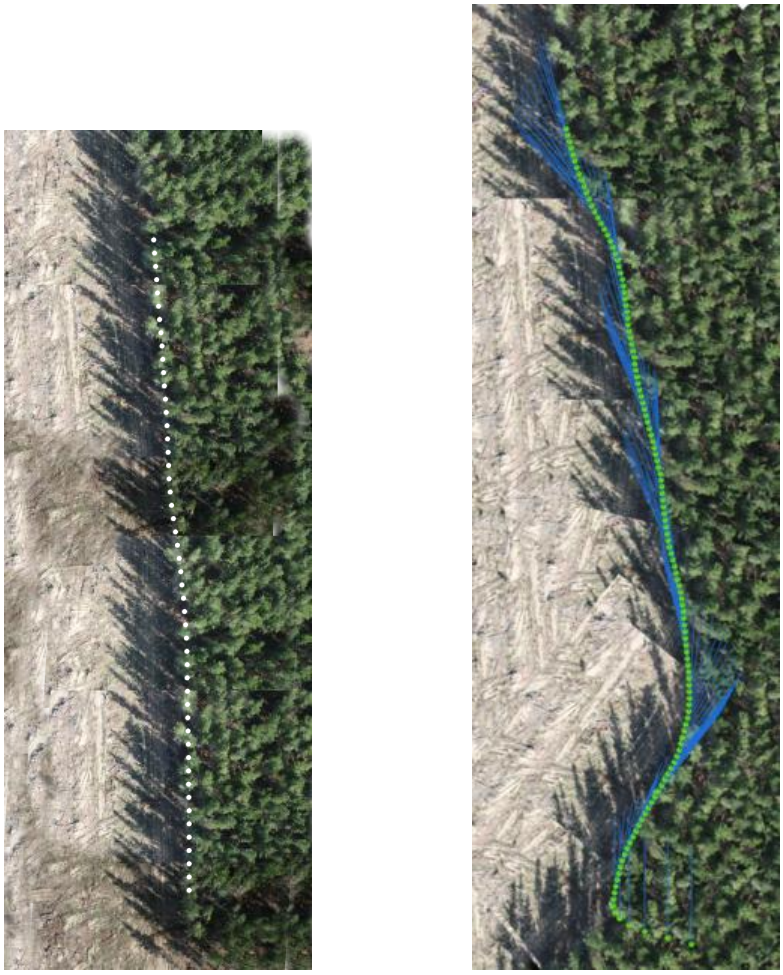
5.2 FOREST DETECTION AND NAVIGATION TESTING

5.2.1 Software Simulation

To assess the performance of the forest edge detection and navigation methods developed, several simulations were performed at different stages of the development process. The first simulation used a composite image produced in image editing software to simulate the length of a cutover edge and was used to test the colour based forest detection method before the SVM classifier was developed. The simulation involved cropping a sub image from the composite image, and treating this as a frame of a video, applying forest edge detection methods, then identifying the direction of the cutover. The output of the navigation algorithm was then used to crop another image in the corresponding direction. The centre of each sub image was plotted onto the composite image, marking the simulated UAV position. This navigation along the cutover edge of the composite image continued until the simulated UAV reached the edge of the image. The initial composite image used in the simulation was quite a simple task for cutover edge detection, despite shadow cast on to the cutover. This is because the edge is mainly straight with only small curves due to tree spacing, there is a no material on the cut edge, and the simulation was started with the center of the first sub image very close to the cutover edge.

In order to test and compare different methods and parameters used for forest detection such as median filter kernel size, image resolution, convex hull, etcetera, a more challenging composite image was made. The second image used the same source images however involved more angular variation making for a more difficult navigation task. Rotation was also added to the simulation so that the detection and response to angular error could be tested. The yaw was displayed on the output by drawing a blue line from the current position in the direction the simulated UAV was heading. Figure 5.3 shows two iterations of the software simulation. Note at the bottom of Figure 5.3b the rapid movement towards the cutover edge.

Thee output of the second simulation was used to asses the accuracy of cutover edge identification and tracking. A hand drawn line was used as ground truth. Figure 5.4a shows the ground truth in green, and the simulation output in red. A python script was used to calculate the discrepancy between the ground truth and the output, giving a mean error of **26.2 pixels**. The window width used for this simulation was 800 pixels, meaning the mean error is only 3.3% of the window width. Based on the size of tree crowns and tree trunks in the image, there are approximately 11 pixels per metre in this test image. Knowing the approximate pixel size the average error in pixels can be



(a) (b)

Figure 5.3 Example software simulation outputs used for testing and development of cutover edge detection algorithms. (a) shows an early example and (b) shows a later example with a more complex edge and yaw control added. The blue lines in the image represent the UAV's heading at a given point.

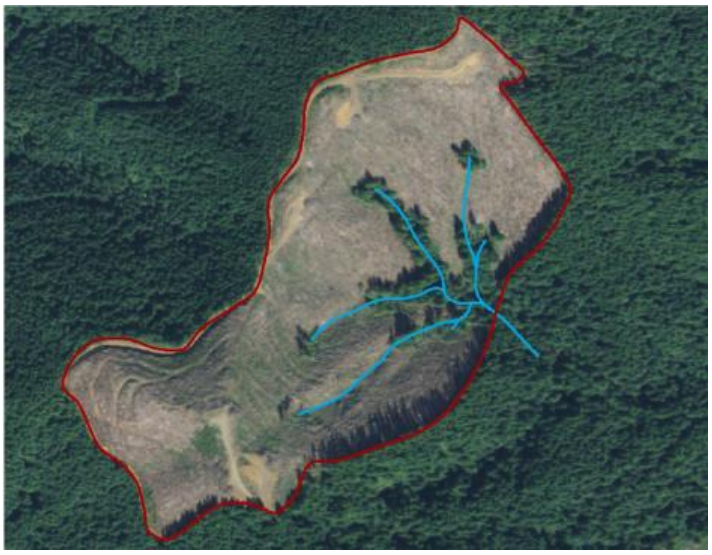
converted into metric units:

$$average metric\, error = 26.2\, pixels \div 11 \frac{pixels}{metre} \tag{5.1}$$

$$= 2.4\, metres \tag{5.2}$$

Compared to hand drawn ground truth, the simulation output lacks precise detail but is much smoother; small scale variations in the ground truth line due to individual trees are not visible. The output is smooth as the detection algorithm examines the area at and ahead of the UAV's current position, and fits a linear line to the detected edge in that region. The smoothing of the small scale features is the reason for the mean error value of 26.2 pixels. While this value is non-trivial, it is considered acceptable as the line still provides an accurate summary of the edge information and is in fact more similar to the type of line drawn in a conventional cutover edge example (Figure 5.4b)

(a) Simulation output (Red) from Figure 5.3b with the ground truth (Green) super imposed.

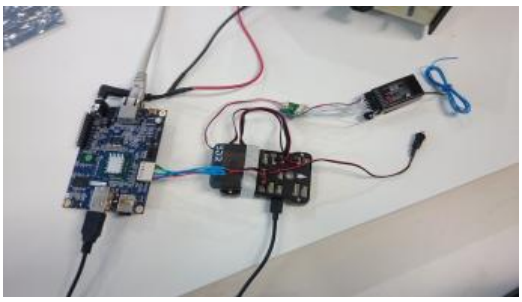


(b) An example provided by SCION of manual cutover edge identification from satellite imagery.

Figure 5.4 Cutover edge classification output from simulation, and an example from satellite imagery.

than the ground truth is. In reality, what actually constitutes the cutover edge is not clearly defined, the full shape of individual trees crowns on the edge could technically be called the cutover edge however such details are generally inconsequential. This is demonstrated clearly in an example of manual cutover edge identification in Figure 5.4b where a smooth line has been drawn and small scale details such as the shape of individual tree crowns ignored. Comparison of the two images in Section 5.2.1 also highlights the significantly higher resolution possible with low altitude imagery, and the associated potential for more accurate edge identification.

It is difficult the accuracy of the simulated UAV in this simulation to prior literature due to the specific nature of the task. A similar example has been identified in Rathinam et al. [2007] where a fixed wing UAV was used to follow the centre of a river. The simulated mean accuracy of 2.4 m in this test compares favorably to the mean value of 7 m in an actual flight test performed in the cited example.



(a) The Minnowboard computer, connected to both the Pixhawk flight computer and the simulation computer.

(b) Groundstation on left display and simulation on right display. RC transmitter is used for flying simulated UAV is visible.

Figure 5.5 Image showing the hardware setup for HITL simulation.

5.2.2 Hardware-in-the-Loop Simulation

In order to test the output of the navigation output and the communication between the Minnowboard max and the Pixhawk, a hard-in-the-loop simulation described in the Pixhawk developer documentation was performed. The simulation uses a desktop computer (as well as the onboard computer) for sending simulated state data to the Pixhawk. The Pixhawk behaves as it would in a typical flight and sends appropriate control outputs which are fed into the simulation. A HITL simulation is useful as it allows testing of aspects of the system that would otherwise require a complete flight to examine. The HITL simulation proved to be particularly useful for setting up the RC transmitter control switches and testing the off board control activation. An image showing the HITL configuration is shown in Figure 5.5.

The open source autopilot interface code (Meier and Lukaczyk [2012]) activates off board mode immediately after the the program starts. This behaviour was altered so that upon activation the onboard computer begins sending control signals to the Pixhawk; these control signals are ignored, until the user activates off board mode using the RC transmitter. It was confirmed using the HITL simulation that there are two ways of switching back from off board control mode. The pilot may at any moment during offboard mode the offboard mode switch to return to the previous mode, or flip a second switch to activate return to launch (RTL). Flipping the RTL switch overrides any other mode switches due to the software architecture of the control loop, as shown in Figure 5.6. The final way offboard mode can be disabled is automatically by the Pixhawk if it does not receive a control signal for a period of greater than 500 ms. This is a useful feature in the flight control software as it means that an unpredicted edge detection crash will not cause the UAV to drop out of the sky, but rather revert to the previous control method.

Another issue identified in the HITL simulation was the difficulty of sending positional commands at a high frequency (greater than 1 Hz). As the forest detection

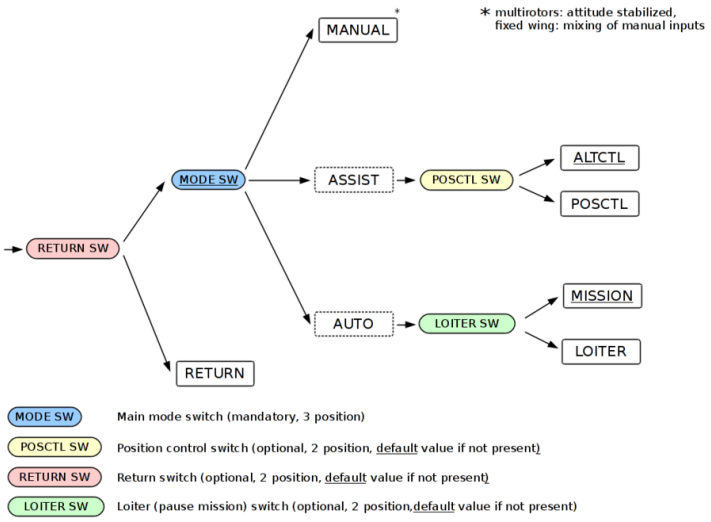


Figure 5.6 System mode architecture for PX4 flight computer (pixhawk.org).

and navigation algorithms run at between 5-15Hz, positional commands are sent to the Pixhawk at the same rate. The Pixhawk begins moving in the right direction but is interrupted by a new command which causes a jerky movement. This is likely why the developer documentation recommends sending velocity commands rather than positional commands when sending directional commands at high frequencies. Velocity set point commands were successfully sent and processed in the HITL test.

5.3 FLIGHT TESTING

After testing verified that the software components were all performing adequately, test flights using the actual quadcopter were performed to test the functionality of the Pixhawk autopilot setup, the onboard control and navigation, and eventually the overall forest cutover following system. Incremental testing of features with increasing complexity was performed to ensure individual aspects were working as expected before proceeding. The tests were set up as follows:

Flight Test 1: The first flight test involved taking the quadcopter for a flight with all supplementary hardware removed to test that the Pixhawk flight controller was behaving as expected and the control layout on the RC transmitter was satisfactory.

Flight Test 2: The second flight test involved adding the Minnowboard computer to the system with a test navigation function which, rather than searching for and following forest, would fly a simple square with sides of 10 m. This served as a test for the navigation system and the communication between the Minnowboard computer and the Pixhawk flight computer during flight. Note that at this stage

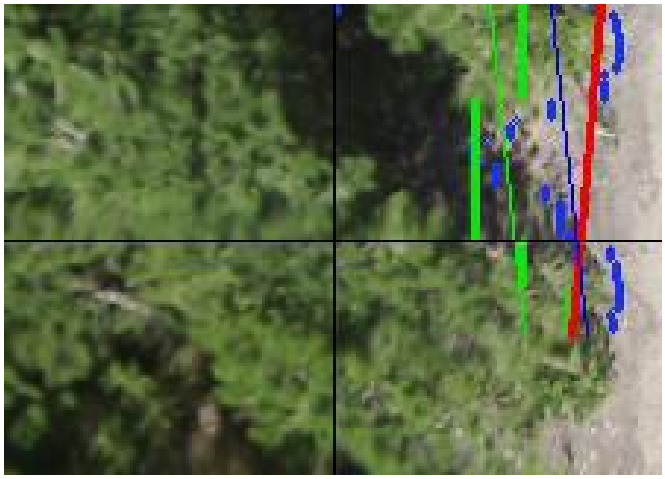


Figure 5.7 A frame from the first Bottle Lake test flight captured and processed during landing. Successful detection of the edge is visible in this frame.

the camera, gimbal, and video transmitter not attached to minimise the potential damage in the event of a crash.

Flight Test 3: In the third flight test, the camera gimbal, camera, and video transmitter were mounted in order to test the quality and range of the video downlink, as well as the functionality of the gimbal.

Flight Test 4: The fourth flight test involved the testing of the complete cutover edge identification and following system, and was performed at Bottle Lake Forest Park. Technical issues relating to the video downlink and Pixhawk firmware meant that a successful test was not performed. Despite this, performance of the proposed cutover edge algorithm was verified, with the forest edge correctly being identified by the SVM and colour selection methods. Section 5.3 shows the output one of these frames.

5.3.1 Bottle Lake Forest Park Test Site

Flight testing of the system over forest was performed with the permission of Rayonier, a global forest products company who are have several sites in the Canterbury region.Sites were recommended by Rayonier in Balmoral forests and Bottle Lake Forest; the latter was chosen as it is much closer to the University of Canterbury. Section 5.3.1 shows a panoramic image of the cutover site, and Section 5.3.1 shows the UAV in flight over trees on the younger side of the forest. The location was appropriate for an initial test flight for the following reasons:

1. The location had clear distinction between cutover and forest,

2. It had young trees on one edge, allowing for testing at a lower altitude than would typically be required for surveying a cutover edge,

3. Despite location in a public park, the area was secluded so there was no risk to other people in the park.

Figure 5.8 A panorama generated of the test flight site. The young forest on the left of the image was used testing the cutover edge detection and navigation.

Figure 5.9 An image take of the UAV in flight at Bottle Lake Forest Park.

5.4 SUMMARY OF RESULTS

The feasibility of cutover edge detection onboard a UAV in real time has been examined and proved in both software simulations and an actual flight test. Further testing is scheduled the system's ability to use the output of the proposed cutover edge identification algorithm to control a UAV by sending navigational commands from the external computer to the flight controller.

Software testing of the system revealed strong performance of the two proposed forest detection algorithms. A pixel wise classification test showed the SVM method performed with an accuracy of 90.2%, and the colour selection method performed with an accuracy of 83.2%. Due to the novel nature of this system, comparison of

results with prior research is tenuous; however the results compare favorably with prior work on image classification using SVM on satellite imagery. Software testing of the proposed combined cutover edge detection algorithm showed promising accuracy with the simulation producing a mean error of 26.2 pixels, corresponding to an approximate metric value of 2.4 metres. Again, a fair comparison is difficult as this is a simulated figure, however a similar work achieved a mean error of 7 m in an actual flight test. Additionally, it would be reasonable to say that the simulated value of 2.4 m compares favorably to existing manual identification of the cutover edge from satellite imagery, due to the low spatial resolution of the images at 17 metres per pixel.

Chapter 6

CONCLUSIONS & FUTURE WORK

6.1 SUMMARY & KEY ACHIEVEMENTS

A proof of concept unmanned aerial system capable of navigating the edge defined by the boundary of forest and cutover has been developed. Two different techniques were developed to classify regions of an image as forest or not forest. The methods were then used together to detect the cutover edge position in a frame. A proportional control system was used to generate navigational instructions based on the position of the UAV above the cutover edge. Mavlink commands were generated on a small Minnowboard Max computer onboard the UAV and sent to the Pixhawk flight computer. Software and hardware performance tested in at different stages to assess the performance of each facet of the system and its interaction with the rest of the system. Successful performance of the proposed combined cutover edge identification algorithm was demonstrated in simulation with an average error of approximately 2.4 m. The simulated accuracy of this system compares favourably to a similar automated UAV system (Rathinam et al. [2007]) whose average positional error was 7 m.

The key achievement and primary contribution of this thesis is the cutover edge detection algorithm. The algorithm combines a classifier based on colour selection and a support vector machine classifier. The two methods are combined to fit a line to the cutover edge in RGB images. Remote sensing techniques such as those examined in Section 2.3 have previously been used to classify forest from remote sensing imagery however the use of a UAV for live navigation based on forest classification output is novel. Use of low altitude UAV imagery for forest classification is also novel, and as such makes comparison of classification accuracy difficult. Classification accuracy of 90.2% was achieved for the proposed SVM based classifier, and 83.2% for the colour selection classifier. Similar work (using multi spectral Landsat imagery) reached a maximum classification accuracy of 87.9% (Pal and Mather [2005]), higher than the proposed colour selection based method but less than the proposed SVM classifier based method.

Previous UAV work in the forestry industry has involved flying a preprogrammed routes with data captured and analysed after the flight. Increased spatial resolution

of low altitude imagery exaggerates features such as shadows between trees, an issue addressed using a combination of morphological and median filtering. It is hoped that successful cutover edge navigation achieved in this thesis will provide a platform which may be further developed and used to streamline current processes of manual mapping used in the forest industry.

6.2 REVIEW OF PROJECT OBJECTIVES AND SCOPE

The objectives set at the start of start of the project (Section 1.1) have been successfully met. Gathering of sample data following a literature review proved to be a time consuming task due to the number of different camera setups tested. Software testing revealed that a developed cutover edge detection algorithm based on RGB video was successful. Porting the algorithm to an embedded system proved to be fairly straight forward affair due to the choice of the Minnowboard Max; a single board computer capable of running the full desktop edition of Linux. The biggest obstacle faced during the development, software testing, and porting was learning to code in the C++ programming language, and learning to operate in a Linux developing environment. Testing of the navigation and cutover edge detection was tested in a hardware in the loop test, a useful tool for verifying the outputs of system match those expected. Flight trials of the system were finally performed, with varying levels of success. Aspects of the system were tested cumulatively: first the UAV was flown to ensure proper operation of the flight computer, then the single board computer was attached and test square flown to verify the onboard computer - flight computer link. All associated video equipment was then attached and tested in flight. Finally the system was tested as a whole with the proposed cutover edge identification algorithm successfully identifying the forest edge in flight. Technical issues with the system delayed testing of the final aspect - using the output of the edge detection to control the UAV autonomously. Unfortunately due to time constraints, the outcome of the final test flight was not able to be included in this thesis.

6.3 FUTURE WORK

Future work on this project will further build on the achievements of this project. It has been shown that the proposed combined SVM and colour selection method of cutover edge identification can successfully be used in flight to linearly approximate a cutover edge. Due to technical issues relating to the onboard computer which runs the proposed edge detection and navigation algorithms, final testing of the system was delayed until after the thesis submission date. It is hoped that the testing will again demonstrate the systems ability to detect the cutover edge onboard the UAV, and will additionally show that this information can be used to navigate the UAV along the cutover edge.

While the proposed detection algorithms have demonstrated feasibility of autonomously detecting the cutover edge, the testing of the system was in clear conditions and a straight, level cutover edge. Further development of the system is required to ensure safe and effective operation of the system, in the range of conditions likely to be encountered in actual forestry operations.

6.3.1 Altitude Management

One particularly important feature of this system is its requirement to maintain a safe operating height above the forest canopy. Maintaining a constant, safe height was not explored in depth on this project as testing was limited to flat areas, however forestry environments are often in hilly or mountainous terrain as is visible in Section 6.3.1.

It was decided early on in the project that a rangefinder would be essential to ensure that the UAV maintained constant clearance above the trees on the cutover edge. Such a rangefinder would hopefully allow the UAV follow height variations in hilly or even mountainous forestry locations, all the while not being affected by speed, wind, noise, ambient light temperature . Using a laser rangefinder was discussed and thought to be a good option due to its associated high accuracy of readings and the high data rate possible.

It is anticipated that with a single laser range finder will be challenging as proposed approach is for the UAV to navigate directly above the height discontinuity that is the cutover edge. With the laser altimeter positioned orthogonally to the ground plane, readings may be of the cutover or of the trees, making it difficult to maintain a constant height. A possible solution which has been discussed but not investigated is use of additional laser range finders. With two laser range finders mounted to the UAV, one could be angled towards each side of the edge so that the height above both ground and forest canopy are known.

A scanning laser range finder (or one dimensional LIDAR) would be optimal as in addition to ensuring a safe height above the forest it would allow for localisation of the edge based on difference in height between the cutover and the forest. Unfortunately a scanning laser range finder suitable for this project in terms of mass, maximum range, and cost was not able to be found. This presents a potential option for future development of the platform as laser devices become more accessible and common within the UAV community.

6.3.2 Safety and Robustness

To ensure effective and safe operation of the system, the robustness of the system's cutover edge, and navigation system needs to be further examined when operating in different environments or conditions. The testing undertaken during this thesis was

Figure 6.1 An image provided by SCION showing a forestry operation on a hill.

limited to flights over a flat cutover in good lighting conditions. In operation the system will be required to operated in a range of different environments. As has been previously noted, forestry operations are located in rugged terrain where the detection task could be complicated by shadows cast over the cutover by sudden drops or rises in the terrain. Future work should consider the following aspects for testing and development of the system:

- Lighting conditions, e.g. solar angle, level of cloud cover, time of day, shading due to geographical features.
- Tree spacing. Greater tree separation will lead to larger shadows between trees, and hence a different texture in captured images
- Presence of unexpected objects, e.g. roads, boulders, dead trees.

Should the proposed cutover edge detection method prove insufficient for operating in across a range of the above aspects then there are several potential avenues which could be explored: The SVM classifier could be retrained to better reflect the actual range in cutover edge conditions. Currently the SVM classifier uses the mean and standard deviation of each colour channel, additional descriptors may be useful for robust classification of forest.

In addition to the performance of the cutover edge detection, the navigation algorithm and its interface with the flight computer should be further tested to ensure the UAV performs consistant and sensible movements. Further testing should consider the behaviour of the system in unexpected events such as a hardware failure. Unit tests have not been developed for this project however would be a valuable way of ensuring the robustness of the navigation and detection algorithms.

6.3.3 Higher Level Navigational Planning

A feature that would be beneficial and required for operation beyond line of site would be the systems ability to make higher level navigational decisions. Currently the system is capable of making low level decisions - move left/right to maintain position above edge, slow down as forest is lost, speed up as forest is found. Examples of more advanced features might be:

- If forest is lost, loop back to last known cutover edge location and try again, if not found there then return home.
- Navigate to a known GPS location on or close to the start of the edge. Begin a search pattern to find the edge.
- Allow input of a GPS finish location with threshold so that when the UAV reaches that location it automatically returns home.
- If an abnormality such as a diseased tree is found then capture its location information and with images for further analysis upon return.

What sets these additional features apart from the current capabilities are their use of additional information to provide higher level operation similar to that of a human pilot. Such features further reduce the required interaction from an operator; eventually a pilot may be redundant (short of transporting the UAV to a suitable take-off location!). Increased autonomy of the system does carry increased liability; if a trained UAV pilot is not closely monitoring the flight then robustness of the system to a range of conditions is of increased importance.

6.3.4 Beyond Line of Sight

In future, testing of the system will require operation of the UAV beyond line of sight. This means that under current CAA regulations, the operation will not fall under part 101 but rather part 102, which requires considerably more regulation of the activity. The legal framework surrounding is likely to change over the course of the next few years however it is essential that future work on this project is carried out with an awareness of current law. Appendix A summarises the current legal framework and implications.

Appendix A

CURRENT LEGAL FRAMEWORK FOR OPERATION OF UAVS IN NEW ZEALAND

Due to the recent increase in use of UAVs, New Zealand's organisation for civil aviation safety and security standards, the CAA, have recently restructured the law surrounding operation of what they describe as remotely piloted aircraft systems (RPAS). The term RPAS is used due to the variety of devices covered by the law, which includes moored balloons and kites, free balloons, rockets, remotely piloted aircraft, control line model aircraft, free flight model aircraft, and gyrogliders and parasails. Under law, operation of RPAS is required to be conducted under either Part 101 or Part 102. It is worth noting that unlike overseas legislation which separates commercial and recreational activities, the CAA has maintained the approach of separating law involving RPAS purely based on the operation itself, rather than its purpose.

A.0.1 Part 101

Operators are not required to seek authorisation of the CAA, as long as the aircraft and operation thereof is in accordance with the requirements of Part 101. There are 12 key features of Part 101 that operators must adhere to:

1. only operate a craft less than 25 kg,
2. take all practicable steps to minimise hazards,
3. only fly during daylight,
4. give way to all crewed craft,
5. be able to see the aircraft with your own eyes to ensure separation from other aircraft,
6. not fly above 120 metres (unless some conditions are met),
7. have knowledge of relevant airspace restrictions,

8. not fly within 4 km of an airport (unless some conditions are met),
9. acquire clearance when operating in controlled airspace ,
10. avoid special-use airspace unless permission has been granted,
11. have consent from anyone you want to fly above,
12. have consent of property owner or person in charge of area you want to fly above.

Aspects 11 and 12 are designed to help protect the safety and privacy of those below the aircraft but have been controversial among recreational drone operators as they complicate the process of going flying. These aspects are not expected to be of consequence for the final operation of the UAV as the operator will be only flying over land that they are in charge of. Flight testing of the system could be complicated by these aspects as relevant permission must be sought each flight.

A.0.2 Visual Line of Sight

In terms of the longer term future of the platform, when flight testing requires following the cutover edge beyond line of sight, aspect 5 will no longer be satisfied, as such operation would clearly be outside the scope of part 101.209 - “Visual line of sight operation”. Part 101.209 section C states that:

"A person who operates an aircraft to which this rule applies must at all times -
(1) maintain visual line of sight with the aircraft; and
(2) be able to see the surrounding airspace in which the craft is operating; and
(3) operate the aircraft below the cloud base."

A.0.3 Part 102

If operation lies outside of the restrictions in Part 101, then special authorisation is required; the operator should apply for an unmanned aircraft operator certificate for operation under Part 102. Certification is granted on a case by case and requires the applicant providing sufficient information about the planned operation to the extent that the Director is satisfied the operation can be conducted safely. The following are some of the aspects required:

- Identification of any person involved
- Locations to be used
- Details of the craft to be used, including the control system
- A hazard register

- Procedures for reporting information to CAA
- Any operating requirements such as pilot or support crew qualifications

It is clear that the process to achieve certification is fairly extensive (and understandingly so). Future work on the development of a autonomous cutover identification system should not take lightly the step up in complexity when transitioning from within the bounds of Part 101 to Part 102.

Appendix B

PREFLIGHT CHECKS

Before travelling to the site of the flight, ensure the following are completed:

- Calibrate sensors using QGroundControl.
- Ensure cutover edge detection software on the Minnowboard is up to date, if not then rebuild.
- Clear or backup old log files and images from Minnowboard.
- Batteries charged.
- RC transmitter charged.
- Permission from property owners/operators granted to fly,
- All required cables are connected securely to the pixhawk:
  - GPS and i2c — GPS antenna
  - Telem1 — Telemetry radio
  - Telem2 — Minnowboard serial connection
  - Switch — Safety switch
  - Buzzer — Speaker
  - Power — Pixhawk power from battery
  - Serial 4/5 — Laser range finder
- Ensure required hardware and gear is packed:
  - Steadidrone
  - RC transmitter
  - Video transmitter and receiver
  - Telemetry radio transmitter and reciever
  - Minnowboard computer and power cable

  - Keyboard and mouse
  - Ground station laptop
  - Phillips head screwdriver
  - High visibility gear for pilot
  - Fire Extinguisher
  - Hardhats (If going to forestry operation)
  - Spark arrestor (If going to forestry operation with non turbo diesel)

On-site preflight check list:

- Equip relevant safety gear.
- Set up telemetry receiver.
- Turn on RC transmitter.
- Plug in battery
- Connect to the Pixhawk from QGroundControl on the laptop.
- Ensure video transmitter and receiver are powered.
- Ensure Minnowboard computer is powered.
- Use video feed to verify Minnowboard has booted successfully.
- Fly

Appendix C

MINNOWBOARD AND PIXHAWK SETUP

C.1 UBUNTU

Installing Ubuntu on the Minnowboard is a straight foward affair:

- Download 64 bit Desktop Ubuntu image from `http://www.ubuntu.com/download/desktop`.
- Create bootable USB drive and insert into Minnowboard.
- Insert microSD card into Minnowboard.
- Power the Minnowboard (ensure steady 5V 3A supply).
- Press the power button to turn on the Minnowboard. Follow the prompts to enter the boot menu and select the Ubuntu USB drive.
- Follow prompts to install Ubuntu as per a typical Ubuntu installation.

C.2 OPENCV

The following link was particularly useful for installation of OpenCV version 2.4.11: `http://www.samontab.com/web/2014/06/installing-opencv-2-4-9-in-ubuntu-14-04-lts/`.

C.3 STARTUP CONFIGURATION

The Minnowboard has been setup so that upon startup the user is logged in and a startup script is run without requiring the user to enter a password or press any buttons. This means that once the board is powered and switched on the onboard forest detection and navigation program will begin running immediately without any interaction.

To enable autologin enter the following commands into the login screen:

```
$ sudo mkdir /etc/lightdm/lightdm.conf.d
$ sudo nano /etc/lightdm/lightdm.conf.d/50-myconfig.conf
```

This will open the nano text editor, enter the following and then save the file:

```
[SeatDefaults]
autologin-user=USERNAME
```

The following script was saved to in the forest detection folder as startupscript.sh. The script saves the output of the program to a time stamped flight log folder for later review. A file is saved in the tmp folder with the process thread's identification number so that the process can easily be killed if required. The startup script used is as follows:

```
#!/bin/sh -e

LOG_FILE="/var/tmp/‘date␣+%Y-%m-%d_%H-%M‘.log"

echo "" > $LOG_FILE
cd /home/dah134/onboardforestdetection
/home/dah134/onboardforestdetection/mavlink_control -d /dev/ttyS0 > $LOG_FILE 2>&1 &

echo $! > /tmp/mavlink_control.pid

exit 0
```

The below corresponds to the executing of the edge detection and navigation program, and the relevant serial port over which communication will occur:

```
/home/dah134/onboardforestdetection/mavlink_control -d /dev/ttyS0 > $LOG_FILE 2>&1 &
```

C.3.1 Allowing sudo access without password

Acessing the serial port of the Minnowboard normally requires authorisation, the following process was used to allow the startup script to use sudo without requiring a password:

C.4 CONFIGURING SERIAL CONNECTION BETWEEN MINNOWBOARD AND PIXHAWK

C.4.1 Serial Cables

A suitable cable connecting the Pixhawk and the Minnowboard was constructed, matching the Pinouts of each device. The Telem2 port on the Pixhawk, and the serial console UART0 port on the Minowboard was used.
Pixhawk pinout:

Pin	Signal	Volt
1 (RED)	VCC	5
2	TX (OUT)	3.3
3	RX (IN)	3.3
4	CTS (IN)	3.3
5	RTS (OUT)	3.3
6	GND	GND

Minowboard 3-wire Serial pinout:

- /dev/ttyS0
- Baud rate: 115200
- Hardware flow control: NO
- Bits: 8
- Stop bits: 1

Pin	Signal	Volt
1	GND	GND
2	/	/
3	/	/
4	TX	3.3
5	RX	3.3
6	/	/

C.4.2 Software Setup and Connection Testing

Following the successful wiring of a serial cable, the Pixhawk must be setup to send Mavlink messages from the Telem2 port. This is performed by adding/editing the configuration file **etc/extras.txt** on the root directory of Pixhawk's microSD card. Don't panic if the directory or file does not exist, it can be created manually and work fine. Add the following line to the extras.txt file:

```
mavlink start -d /dev/ttyS2 -b 57600
```

The '-d /dev/ttys2' refers to the port (Telem2) on which Mavlink is being started and '-b 57600' refers to the baud rate. After editing the file Pixhawk will now start Mavlink on Telem2 when it boots up. Plug the serial cable into the Telem2 port and onboard computer's serial port, turn on the Pixhawk, and type the following into terminal to test the connection:

```
$ cat /dev/ttyS0
```

The terminal should begin printing garbage characters, use Ctrl + C to quit the terminal immediately. If nothing is printed then you may not have the required permissions to access the serial port. Type the following command into terminal to give you full permissions:

```
$ sudo chmod 666 /dev/ttyS0
```

For more detail see:
`https://pixhawk.org/dev/mavlink_message_passthrough`
`https://pixhawk.org/dev/companion_link`
`https://pixhawk.org/dev/system_startup`

Appendix D

HARDWARE IN THE LOOP SETUP

A HITL testing is a useful tool for assessing the performance of the system without the risk of dropping out of the sky. A computer is used to simulate the inertial and GPS sensor inputs to the Pixhawk. The simulated signals are sent to the Pixhawk flight computer over USB and the Pixhawk's response is returned to a computer (a computer Lwhich simulates the motion of the UAV. HITL installation is easiest on a Linux/Mac machine (tested on Ubuntu).

D.1 INSTALLATION

Install required dependencies:

```
$ sudo apt-get install ant openjdk-7-jdk openjdk-7-jre git-all
```

Copy jmavsim from repository:

```
$ git clone https://github.com/PX4/jMAVSim.git
```

Change directory to the copied jmavsim folder and initialise GIT repository:

```
$ cd jMAVSim
$ git submodule init
$ git submodule update
```

Build:

```
$ ant
```

D.2 PIXHAWK SETUP

To configure the Pixhawk for use in HITL mode, use QGroundControl to change the airframe:

- Connect to the Pixhawk over USB from QgroundControl
- Go to Setup

- Go to Airframe
- Select the relevant HITL airframe type
- Apply and restart. Do NOT reconnect to the Pixhawk

D.3 RUNNING HITL

To run the simulation, complete the following steps in order:

1. Make sure QGroundControl is not connected to the Pixhawk
2. Turn on RC transmitter
3. Run JMavSim by executing the following command, changing the underlined text to the USB connection used for connecting to the Pixhawk:

   ```
   $ java -Djava.ext.dirs= -cp lib/*:out/production/jmavsim.jar
   me.drton.jmavsim.Simulator -serial /dev/ttyACM0 921600 âÃŞqgc
   ```

4. Using QGroundControl connect to the Pixhawk using the default UDP link from the connect button. Messages should begin showing in the JMavSim terminal window.
5. To facilitate viewing in QGroundControl, select the view from down menu and check the HITL mode option.

This section is a combination of the descriptions in online documentation, see the following for greater detail:
`https://pixhawk.org/dev/hil/jmavsim`
`https://pixhawk.org/users/hil`
`https://pixhawk.org/dev/simulation/start`
`https://github.com/DrTon/jMAVSim`

REFERENCES

Allen, C.T. (1995), 'Interferometric synthetic aperture radar', *IEEE Geoscience and Remote Sensing Society Newsletter*, Vol. 96, pp. 6–13.

altus.com (2016), 'New lidar systems', January, [Online; accessed 5-January-2016].

Baker, P. and Kamgar-Parsi, B. (2010), 'Using shorelines for autonomous air vehicle guidance', *Computer Vision and Image Understanding*, Vol. 114, No. 6, pp. 723–729.

Borş, A.G. and Pitas, I. (1998), 'Optical flow estimation and moving object segmentation based on median radial basis function network', *Image Processing, IEEE Transactions on*, Vol. 7, No. 5, pp. 693–702.

Bradski, G. (), *Dr. Dobb's Journal of Software Tools*.

Braillon, C., Pradalier, C., Crowley, J.L. and Laugier, C. (2006), 'Real-time moving obstacle detection using optical flow models', In *Intelligent Vehicles Symposium, 2006 IEEE*, IEEE, pp. 466–471.

Byrne, J., Cosgrove, M. and Mehra, R. (2006), 'Stereo based obstacle detection for an unmanned air vehicle', In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, IEEE, pp. 2830–2835.

Carrillo, L.R.G., López, A.E.D., Lozano, R. and Pégard, C. (2012), 'Combining stereo vision and inertial navigation system for a quad-rotor uav', *Journal of Intelligent & Robotic Systems*, Vol. 65, No. 1-4, pp. 373–387.

Chou, T.Y., Yeh, M.L., Chen, Y.C. and Chen, Y.H. (2010), *Disaster monitoring and management by the unmanned aerial vehicle technology*, na.

coptercraft.com (2016), 'Aerial uav lidar systems', January, [Online; accessed 3-January-2016].

Corona, P., Cartisano, R., Salvati, R., Chirici, G., Floris, A., Di Martino, P., Marchetti, M., Scrinzi, G., Clementel, F. and Torresan, C. (2012), 'Airborne laser scanning to support forest resource management under alpine, temperate and mediterranean environments in italy', .

Dickmanns, E.D. and Mysliwetz, B.D. (1992), 'Recursive 3-d road and relative ego-state recognition', *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , No. 2, pp. 199–213.

Dinuls, R., Erins, G., Lorencs, A., Mednieks, I. and Sinica-Sinavskis, J. (2012), 'Tree species identification in mixed baltic forest using lidar and multispectral data', *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, Vol. 5, No. 2, pp. 594–603.

Domínguez-Morales, M., Jiménez-Fernández, A., Linares-Barranco, A., Jiménez-Moreno, G. and Paz-Vicente, R. (2012), *Stereo matching: From the basis to neuromorphic engineering*, INTECH Open Access Publisher.

Eisenbeiss, H., Lambers, K. and Sauerbier, M. (2005), *Photogrammetric recording of the archaeological site of Pinchango Alto (Palpa, Peru) using a mini helicopter (UAV)*, Bibliothek der Universität Konstanz.

Esposito a, S., Murab, M., Fallavollitaa, P., Balsia, M., Chiricib, G., Oradinic, A. and MarchettiZ, M. (2014), 'Performance evaluation of lightweight lidar for uav applications', In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, IEEE, pp. 792–795.

Farnebäck, G. (2003), 'Two-frame motion estimation based on polynomial expansion', In *Image Analysis*, pp. 363–370, Springer.

Franklin, S. and McDermid, G. (1993), 'Empirical relations between digital spot hrv and casi spectral response and lodgepole pine (pinus contorta) forest stand parameters', *International Journal of Remote Sensing*, Vol. 14, No. 12, pp. 2331–2348.

Franklin, S. et al. (2001), 'Remote sensing for sustainable forest management.', *Remote sensing for sustainable forest management*.

Frew, E., McGee, T., Kim, Z., Xiao, X., Jackson, S., Morimoto, M., Rathinam, S., Padial, J. and Sengupta, R. (2004), 'Vision-based road-following using a small autonomous aircraft', In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, Vol. 5, IEEE, pp. 3006–3015.

Graham, B. (2015), Personal communication, February.

He, K., Zhang, X., Ren, S. and Sun, J. (2015), 'Delving deep into rectifiers: Surpassing human-level performance on imagenet classification', *arXiv preprint arXiv:1502.01852*.

Herisse, B., Russotto, F.X., Hamel, T. and Mahony, R. (2008), 'Hovering flight and vertical landing control of a vtol unmanned aerial vehicle using optical flow', In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, IEEE, pp. 801–806.

Horn, B.K. and Schunck, B.G. (1981), 'Determining optical flow', In *1981 Technical symposium east*, International Society for Optics and Photonics, pp. 319–331.

Howard, I.P. and Rogers, B.J. (1995), *Binocular vision and stereopsis*, Oxford University Press.

Hrabar, S. (2008), '3d path planning and stereo-based obstacle avoidance for rotorcraft uavs', In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, IEEE, pp. 807–814.

Hrabar, S., Sukhatme, G.S., Corke, P., Usher, K. and Roberts, J. (2005), 'Combined optic-flow and stereo-based navigation of urban canyons for a uav', In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, IEEE, pp. 3309–3316.

Hsu, C.W., Chang, C.C., Lin, C.J. et al. (2003), 'A practical guide to support vector classification', .

Hummel, S., Hudak, A., Uebler, E., Falkowski, M. and Megown, K. (2011), 'A comparison of accuracy and cost of lidar versus stand exam data for landscape management on the malheur national forest', *Journal of forestry*, Vol. 109, No. 5, pp. 267–273.

Korosec, K. (2015), 'Elon musk says tesla vehicles will drive themselves in two years', December, [Online; posted 21-December-2015].

Krajník, T., Nitsche, M., Pedre, S., Přeučil, L. and Mejail, M.E. (2012), 'A simple visual navigation system for an uav', In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, IEEE, pp. 1–6.

Lane, C.R., Liu, H., Autrey, B.C., Anenkhonov, O.A., Chepinoga, V.V. and Wu, Q. (2014), 'Improved wetland classification using eight-band high resolution satellite imagery and a hybrid approach', *Remote Sensing*, Vol. 6, No. 12, pp. 12187–12216.

Lim, K., Treitz, P., Wulder, M., St-Onge, B. and Flood, M. (2003), 'Lidar remote sensing of forest structure', *Progress in physical geography*, Vol. 27, No. 1, pp. 88–106.

Lucas, B.D., Kanade, T. et al. (1981), 'An iterative image registration technique with an application to stereo vision.', In *IJCAI*, Vol. 81, pp. 674–679.

Lucieer, A., de Jong, S. and Turner, D. (2013), 'Mapping landslide displacements using structure from motion (sfm) and image correlation of multi-temporal uav photography', *Progress in Physical Geography*, p. 0309133313515293.

Mae, Y., Shirai, Y., Miura, J. and Kuno, Y. (1996), 'Object tracking in cluttered background based on optical flow and edges', In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, Vol. 1, IEEE, pp. 196–200.

Massonnet, D. and Feigl, K.L. (1998), 'Radar interferometry and its application to changes in the earth's surface', *REVIEWS OF GEOPHYSICS-RICHMOND VIRGINIA THEN WASHINGTON-*, Vol. 36, pp. 441–500.

Matthies, L., Kelly, A., Litwin, T. and Tharp, G. (1996), *Obstacle detection for unmanned ground vehicles: A progress report*, Springer.

Meier, L. and Lukaczyk, T. (2012), 'Mavlink, c uart interface example', `https://github.com/mavlink/c_uart_interface_example`.

Merrell, P.C., Lee, D.J. and Beard, R.W. (2004), 'Obstacle avoidance for unmanned air vehicles using optical flow probability distributions', In *Optics East*, International Society for Optics and Photonics, pp. 13–22.

Montaghi, A., Corona, P., Dalponte, M., Gianelle, D., Chirici, G. and Olsson, H. (2013), 'Airborne laser scanning of forest resources: an overview of research in italy as a commentary case study', *International Journal of Applied Earth Observation and Geoinformation*, Vol. 23, pp. 288–300.

mosaicmill.com (2015), 'Mosaicmill forestry applications', January, [Online; accessed 02-January-2016].

Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F. and Sayd, P. (2009), 'Generic and real-time structure from motion using local bundle adjustment', *Image and Vision Computing*, Vol. 27, No. 8, pp. 1178–1193.

OxfordDictionary (2015), *Oxford Dictionary of English*, Oxford University Press.

Pal, M. and Mather, P. (2005), 'Support vector machines for classification in remote sensing', *International Journal of Remote Sensing*, Vol. 26, No. 5, pp. 1007–1011.

Pinz, A.J. (1991), 'A computer vision system for the recognition of trees in aerial photographs', .

pixhawk.org (), 'System modes', [Online; accessed 12-November-2019].

Pratt, K., Murphy, R., Stover, S. and Griffin, C. (2006), 'Requirements for semi-autonomous flight in miniature uavs for structural inspection', *AUVSI's Unmanned Systems North America. Orlando, Florida, Association for Unmanned Vehicle Systems International*.

Rankin, A., Huertas, A. and Matthies, L. (2005), *Evaluation of stereo vision obstacle detection algorithms for off-road autonomous navigation*, Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration.

Rathinam, S., Almeida, P., Kim, Z., Jackson, S., Tinka, A., Grossman, W. and Sengupta, R. (2007), 'Autonomous searching and tracking of a river using an uav', In *American Control Conference, 2007. ACC'07*, IEEE, pp. 359–364.

Riegl (), 'Riegl vux-1uav', .

Romero, H., Salazar, S. and Lozano, R. (2009), 'Real-time stabilization of an eight-rotor uav using optical flow', *Robotics, IEEE Transactions on*, Vol. 25, No. 4, pp. 809–817.

Routscene (), 'Uav lidarpod', .

Saripalli, S., Montgomery, J.E. and Sukhatme, G.S. (2002), 'Vision-based autonomous landing of an unmanned aerial vehicle', In *Robotics and automation, 2002. Proceedings. ICRA'02. IEEE international conference on*, Vol. 3, IEEE, pp. 2799–2804.

Scharstein, D. and Szeliski, R. (2003), 'High-accuracy stereo depth maps using structured light', In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, Vol. 1, IEEE, pp. I–195.

Sezgin, M. et al. (2004), 'Survey over image thresholding techniques and quantitative performance evaluation', *Journal of Electronic imaging*, Vol. 13, No. 1, pp. 146–168.

Shakernia, O., Vidal, R., Sharp, C.S., Ma, Y. and Sastry, S. (2002), 'Multiple view motion estimation and control for landing an unmanned aerial vehicle', In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, Vol. 3, IEEE, pp. 2793–2798.

smartplanes.com (2015), 'Smartplanes forestry', January, [Online; accessed 02-January-2016].

Sonka, M., Hlavac, V. and Boyle, R. (2014), *Image processing, analysis, and machine vision*, Cengage Learning.

steadidrone.com (2015), 'Steadidrone models', [Online; accessed 4-January-2016].

Stefanik, K.V., Gassaway, J.C., Kochersberger, K. and Abbott, A.L. (2011), 'Uav-based stereo vision for rapid aerial terrain mapping', *GIScience & Remote Sensing*, Vol. 48, No. 1, pp. 24–49.

Strydom, R., Thurrowgood, S. and Srinivasan, M. (2014), 'Visual odometry: autonomous uav navigation using optic flow and stereo', In *Australasian Conf. on Robotics and Automation (ACRA)(Melbourne)*, .

Survey, U. (2015), 'Landsat 8', January, [Online; edited 12-January-2015].

Taigman, Y., Yang, M., Ranzato, M. and Wolf, L. (2014), 'Deepface: Closing the gap to human-level performance in face verification', In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, IEEE, pp. 1701–1708.

UAV, V. (), 'Lidar mapping', .

Ullman, S. (1979), 'The interpretation of structure from motion', *Proceedings of the Royal Society of London B: Biological Sciences*, Vol. 203, No. 1153, pp. 405–426.

Van Der Mark, W., Groen, F. and Van Den Heuvel, J. (2001), 'Stereo based navigation in unstructured environments', In *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, Vol. 3, IEEE, pp. 2038–2043.

Velodyne (2015), 'Velodyne lidar puck', [Online; accessed 02-January-2016].

Vohland, M., Stoffels, J., Hau, C. and Schuler, G. (2007), 'Remote sensing techniques for forest parameter assessment: multispectral classification and linear spectral mixture analysis', *Silva Fennica*, Vol. 41, No. 3, p. 441.

Wallace, L., Lucieer, A., Watson, C. and Turner, D. (2012), 'Development of a uav-lidar system with application to forest inventory', *Remote Sensing*, Vol. 4, No. 6, pp. 1519–1543.

Wanless, B. (2013), 'Canadian uav company maps forest fire damage in alberta', *GoGeomatics Canada Magazine*.

Webster, G. (2013), 'Nasa's mars curiosity debuts autonomous navigation', August, [Online; posted 27-August-2013].

wikipedia.com (2010), 'Hsv color solid cylinder', March, [Online; posted 22-March-2010].

Wulder, M.A., Bater, C.W., Coops, N.C., Hilker, T. and White, J.C. (2008), 'The role of lidar in sustainable forest management', *The Forestry Chronicle*, Vol. 84, No. 6, pp. 807–826.

XactSense (), 'Custom uav for lidar survey and mapping', .

Yang, L., Wu, X., Praun, E. and Ma, X. (2009), 'Tree detection from aerial imagery', In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, pp. 131–137.